# Generative Models for Statistical Parsing with Combinatory Categorial Grammar

**Julia Hockenmaier** and **Mark Steedman**
Division of Informatics
University of Edinburgh
Edinburgh EH8 9LW, United Kingdom
{julia, steedman}@cogsci.ed.ac.uk

## Abstract

This paper compares a number of generative probability models for a wide-coverage Combinatory Categorial Grammar (CCG) parser. These models are trained and tested on a corpus obtained by translating the Penn Treebank trees into CCG normal-form derivations. According to an evaluation of unlabeled word-word dependencies, our best model achieves a performance of 89.9%, comparable to the figures given by Collins (1999) for a linguistically less expressive grammar. In contrast to Gildea (2001), we find a significant improvement from modeling word-word dependencies.

## 1 Introduction

The currently best single-model statistical parser (Charniak, 1999) achieves Parseval scores of over 89% on the Penn Treebank. However, the grammar underlying the Penn Treebank is very permissive, and a parser can do well on the standard Parseval measures without committing itself on certain semantically significant decisions, such as predicting null elements arising from deletion or movement. The potential benefit of wide-coverage parsing with CCG lies in its more constrained grammar and its simple and semantically transparent capture of extraction and coordination.

We present a number of models over syntactic derivations of Combinatory Categorial Grammar (CCG, see Steedman (2000) and Clark et al. (2002), this conference, for introduction), estimated from and tested on a translation of the Penn Treebank to a corpus of CCG normal-form derivations. CCG grammars are characterized by much larger category sets than standard Penn Treebank grammars, distinguishing for example between many classes of verbs with different subcategorization frames. As a result, the categorial lexicon extracted for this purpose from the training corpus has 1207 categories, compared with the 48 POS-tags of the Penn Treebank. On the other hand, grammar rules in CCG are limited to a small number of simple unary and binary combinatory schemata such as function application and composition. This results in a smaller and less overgenerating grammar than standard PCFGs (ca. 3,000 rules when instantiated with the above categories in sections 02-21, instead of >12,400 in the original Treebank representation (Collins, 1999)).

## 2 Evaluating a CCG parser

Since CCG produces unary and binary branching trees with a very fine-grained category set, CCG Parseval scores cannot be compared with scores of standard Treebank parsers. Therefore, we also evaluate performance using a dependency evaluation reported by Collins (1999), which counts word-word dependencies as determined by local trees and their labels. According to this metric, a local tree with parent node $P$, head daughter $H$ and non-head daughter $S$ (and position of $S$ relative to $P$, ie. left or right, which is implicit in CCG categories) defines a $\langle P, H, S\rangle$ dependency between the head word of $S$, $w_S$, and the head word of $H$, $w_H$. This measure is neutral with respect to the branching factor. Furthermore, as noted by Hockenmaier (2001), it does not penalize equivalent analyses of multiple modi-

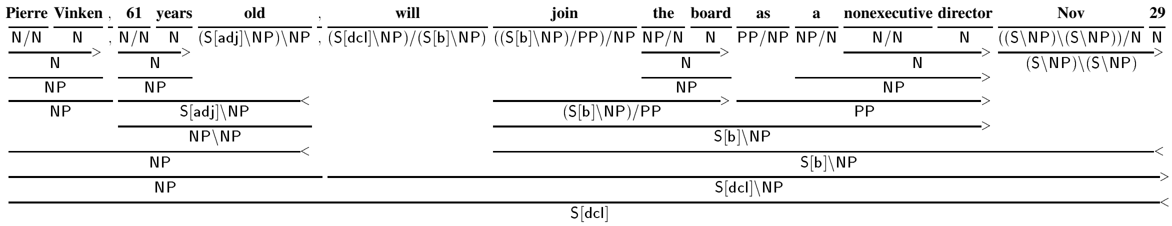



Figure 1: A CCG derivation in our corpus

fiers. In the unlabeled case $\langle\rangle$ (where it only matters whether word *a* is a dependent of word *b*, not what the label of the local tree is which defines this dependency), scores can be compared across grammars with different sets of labels and different kinds of trees. In order to compare our performance with the parser of Clark et al. (2002), we also evaluate our best model according to the dependency evaluation introduced for that parser. For further discussion we refer the reader to Clark and Hockenmaier (2002) .

## 3 CCGbank—a CCG treebank

CCGbank is a corpus of CCG normal-form derivations obtained by translating the Penn Treebank trees using an algorithm described by Hockenmaier and Steedman (2002). Almost all types of construction—with the exception of gapping and UCP ("Unlike Coordinate Phrases") are covered by the translation procedure, which processes 98.3% of the sentences in the training corpus (WSJ sections 02-21) and 98.5% of the sentences in the test corpus (WSJ section 23). The grammar contains a set of type-changing rules similar to the lexical rules described in Carpenter (1992). Figure 1 shows a derivation taken from CCGbank. Categories, such as ((S[b]\NP)/PP)/NP, encode unsaturated subcat frames. The complement-adjunct distinction is made explicit; for instance *as a nonexecutive director* is marked up as PP−CLR in the Treebank, and hence treated as a PP-complement of *join*, whereas *Nov. 29* is marked up as an NP−TMP and therefore analyzed as VP modifier. The −CLR tag is not in fact a very reliable indicator of whether a constituent should be treated as a complement, but the translation to CCG is automatic and must do the best it can with the information in the Treebank.

The verbal categories in CCGbank carry features distinguishing declarative verbs (and auxiliaries) from past participles in past tense, past participles for passive, bare infinitives and ing-forms. There is a separate level for nouns and noun phrases, but, like the nonterminal NP in the Penn Treebank, noun phrases do not carry any number agreement. The derivations in CCGbank are "normal-form" in the sense that analyses involving the combinatory rules of type-raising and composition are only used when syntactically necessary.

## 4 Generative models of CCG derivations

| | Expansion $P(exp \mid \ldots)$ | HeadCat $P(H \mid \ldots)$ | NonHeadCat $P(S \mid \ldots)$ |
|---|---|---|---|
| Baseline | $P$ | $P, exp$ | $P, exp, H$ |
| + Conj | $P, conj_P$ | $P, exp, conj_P$ | $P, exp, H, conj_P$ |
| + Grandparent | $P, GP$ | $P, GP, exp$ | $P, GP, exp, H$ |
| + $\Delta$ | $P\#\Delta^{L,R}{}_P$ | $P, exp\#\Delta^{L,R}{}_P$ | $P, exp, H\#\Delta^{L,R}{}_P$ |

Table 1: The unlexicalized models

The models described here are all extensions of a very simple model which models derivations by a top-down tree-generating process. This model was originally described in Hockenmaier (2001), where it was applied to a preliminary version of CCGbank, and its definition is repeated here in the top row of Table 1. Given a (parent) node with category *P*, choose the **expansion** *exp* of *P*, where *exp* can be leaf (for lexical categories), unary (for unary expansions such as type-raising), left (for binary trees where the head daughter is left) or right (binary trees, head right). If *P* is a leaf node, generate its **head word** *w*. Otherwise, generate the category of its **head daughter** *H*. If *P* is binary branching, generate the category of its **non-head daughter** *S* (a complement or modifier of *H*).

The model itself includes no prior knowledge specific to CCG other than that it only allows unary and binary branching trees, and that the sets of nonterminals and terminals are not disjoint (hence the need to include leaf as a possible expansion, which acts as a stop probability).

All the experiments reported in this section were conducted using sections 02-21 of CCGbank as training corpus, and section 23 as test corpus. We replace all rare words in the training data with their POS-tag. For all experiments reported here and in section 5, the frequency threshold was set to 5. Like Collins (1999), we assume that the test data is POS-tagged, and can therefore replace unknown words in the test data with their POS-tag, which is more appropriate for a formalism like CCG with a large set of lexical categories than one generic token for all unknown words.

The performance of the baseline model is shown in the top row of table 3. For six out of the 2379 sentences in our test corpus we do not get a parse.[1] The reason is that a lexicon consisting of the word-category pairs observed in the training corpus does not contain all the entries required to parse the test corpus. We discuss a simple, but imperfect, solution to this problem in section 7.

## 5 Extending the baseline model

State-of-the-art statistical parsers use many other features, or conditioning variables, such as head words, subcategorization frames, distance measures and grandparent nodes. We too can extend the baseline model described in the previous section by including more features. Like the models of Goodman (1997), the additional features in our model are generated probabilistically, whereas in the parser of Collins (1997) distance measures are assumed to be a function of the already generated structure and are not generated explicitly.

In order to estimate the conditional probabilities of our model, we recursively smooth empirical estimates $\hat{e}_i$ of specific conditional distributions with (possible smoothed) estimates of less specific distributions $\tilde{e}_{i-1}$, using linear interpolation:

$$\tilde{e}_i = \lambda \hat{e}_i + (1-\lambda)\tilde{e}_{i-1}$$

$\lambda$ is a smoothing weight which depends on the particular distribution.[2]

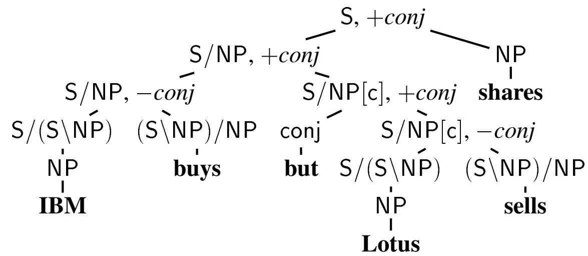
When defining models, we will indicate a back-off level with a # sign between conditioning variables, eg. $A,B$ # $C$ # $D$ means that we interpolate $\hat{P}(\ldots \mid A,B,C,D)$ with $\tilde{P}(\ldots \mid A,B,C)$, which is an interpolation of $\hat{P}(\ldots \mid A,B,C)$ and $\hat{P}(\ldots \mid A,B)$.

[1] We conjecture that the minor variations in coverage among the other models (except Grandparent) are artefacts of the beam.

[2] We compute $\lambda$ in the same way as Collins (1999), p. 185.

### 5.1 Adding non-lexical information

**The coordination feature** We define a boolean feature, *conj*, which is true for constituents which expand to coordinations on the head path.




This feature is generated at the root of the sentence with $P(conj \mid \mathrm{TOP})$. For binary expansions, $conj_H$ is generated with $P(conj_H \mid H, S, conj_P)$ and $conj_S$ is generated with $P(conj_S \mid S \# P, exp_P, H, conj_P)$. Table 1 shows how *conj* is used as a conditioning variable. This is intended to allow the model to capture the fact that, for a sentence without extraction, a CCG derivation where the subject is type-raised and composed with the verb is much more likely in right node raising constructions like the above.

**The impact of the grandparent feature** Johnson (1998) showed that a PCFG estimated from a version of the Penn Treebank in which the label of a node's parent is attached to the node's own label yields a substantial improvement (LP/LR: from 73.5%/69.7% to 80.0%/79.2%). The inclusion of an additional grandparent feature gives Charniak (1999) a slight improvement in the Maximum Entropy inspired model, but a slight decrease in performance for an MLE model. Table 3 (**Grandparent**) shows that a grammar transformation like Johnson's does yield an improvement, but not as dramatic as in the Treebank-CFG case. At the same time coverage is reduced (which might not be the case if this was an additional feature in the model rather than a change in the representation of the categories). Both of these results are to be expected—CCG categories encode more contextual information than Treebank labels, in particular about parents and grandparents; therefore the history feature might be expected to have less impact. Moreover, since our category set is much larger, appending the parent node will lead to an even more fine-grained partitioning of the data, which then results in sparse data problems.

**Distance measures for CCG** Our distance measures are related to those proposed by Goodman (1997), which are appropriate for binary trees (unlike those of Collins (1997)). Every node has a left distance measure, $\Delta^L$, measuring the distance from the head word to the left frontier of the constituent. There is a similar right distance measure $\Delta^R$. We implemented three different ways of measuring distance: $\Delta_{\text{Adjacency}}$ measures string adjacency (0, 1 or 2 and more intervening words); $\Delta_{\text{Verb}}$ counts intervening verbs (0 or 1 and more); and $\Delta_{\text{Pct}}$ counts intervening punctuation marks (0, 1, 2 or 3 and more). These $\Delta$s are generated by the model in the following manner: at the root of the sentence, generate $\Delta^L$ with $P(\Delta^L \mid \text{TOP})$, and $\Delta^R$ with $P(\Delta^R \mid \text{TOP}, \Delta^L)$. Then, for each expansion, if it is a unary expansion, ${\Delta^L}_H = {\Delta^L}_P$ and ${\Delta^R}_H = {\Delta^R}_P$ with a probability of 1. If it is a binary expansion, only the $\Delta$ in the direction of the sister changes, with a probability of $P({\Delta^L}_H \mid {\Delta^L}_P H\#P, S)$ if $exp = \texttt{right}$, and analogously for $exp = \texttt{left}$. ${\Delta^L}_S$ and ${\Delta^R}_S$ are conditioned on $S$ and the $\Delta$ of $H$ and $P$ in the direction of $S$: $P({\Delta^L}_S \mid S\#{\Delta^R}_P, {\Delta^R}_H)$ and $P({\Delta^R}_S \mid S, {\Delta^L}_S\#{\Delta^R}_P, {\Delta^R}_H)$. They are then used as further conditioning variables for the other distributions as shown in table 1.

Table 3 also gives the Parseval and dependency scores obtained with each of these measures. $\Delta_{\text{Pct}}$ has the smallest effect. However, our model does not yet contain anything like the hard constraint on punctuation marks in Collins (1999).

### 5.2 Adding lexical information

Gildea (2001) shows that removing the lexical dependencies in Model 1 of Collins (1997) (that is, not conditioning on $w_h$ when generating $w_s$) decreases labeled precision and recall by only 0.5%. It can therefore be assumed that the main influence of lexical head features (words and preterminals) in Collins' Model 1 is on the structural probabilities.

In CCG, by contrast, preterminals are lexical categories, encoding complete subcategorization information. They therefore encode more information about the expansion of a nonterminal than Treebank POS-tags and thus are more constraining.

Generating a constituent's lexical category $c$ at its maximal projection (ie. either at the root of the tree, TOP, or when generating a non-head daughter $S$), and using the lexical category as conditioning variable (**LexCat**) increases performance of the baseline model as measured by $\langle P, H, S \rangle$ by almost 3%. In this model, $c_S$, the lexical category of $S$ depends on the category $S$ and on the local tree in which $S$ is generated. However, slightly worse performance is obtained for **LexCatDep**, a model which is identical to the original **LexCat** model, except that $c_S$ is also conditioned on $c_H$, the lexical category of the head node, which introduces a dependency between the lexical categories.

Since there is so much information in the lexical categories, one might expect that this would reduce the effect of conditioning the expansion of a constituent on its head word $w$. However, we did find a substantial effect. Generating the head word at the maximal projection (**HeadWord**) increases performance by a further 2%. Finally, conditioning $w_S$ on $w_H$, hence including word-word dependencies, (**HWDep**) increases performance even more, by another 3.5%, or 8.3% overall. This is in stark contrast to Gildea's findings for Collins' Model 1.

We conjecture that the reason why CCG benefits more from word-word dependencies than Collins' Model 1 is that CCG allows a cleaner parametrization of these surface dependencies. In Collins' Model 1, $w_S$ is conditioned not only on the local tree $\langle P, H, S \rangle$, $c_H$ and $w_H$, but also on the distance $\Delta$ between the head and the modifier to be generated. However, Model 1 does not incorporate the notion of subcategorization frames. Instead, the distance measure was found to yield a good, if imperfect, approximation to subcategorization information.

Using our notation, Collins' Model 1 generates $w_S$ with the following probability:

$$
\begin{aligned}
&P_{Collins1}(w_S \mid c_S, \Delta, P, H, S, c_H, w_H) = \\
&\lambda_1 \hat{P}(w_S \mid c_S, \Delta, P, H, S, c_H, w_H) \\
&+ (1-\lambda_1)\left[\lambda_2 \hat{P}(w_S \mid c_S, \Delta, P, H, S, c_H) + (1-\lambda_2)\hat{P}(w_S \mid c_S)\right]
\end{aligned}
$$

—whereas the CCG dependency model generates $w_S$ as follows:

$$
\begin{aligned}
&P_{CCGdep}(w_S \mid c_S, P, H, S, c_H, w_H) = \\
&\lambda \hat{P}(w_S \mid c_S, P, H, S, c_H, w_H) + (1-\lambda)\hat{P}(w_S \mid c_S)
\end{aligned}
$$

Since our $P$, $H$, $S$ and $c_H$ are CCG categories, and hence encode subcategorization information, the local tree always identifies a specific argument slot. Therefore it is not necessary for us to include a distance measure in the dependency probabilities.

| | Expansion | HeadCat | NonHeadCat | LexCat | | Head word | |
|---|---|---|---|---|---|---|---|
| | $P(exp \mid \ldots)$ | $P(H \mid \ldots)$ | $P(S \mid \ldots)$ | $P(c_S \mid \ldots)$ | $P(c_{\mathrm{TOP}} \mid \ldots)$ | $P(w_S \mid \ldots)$ | $P(w_{\mathrm{TOP}} \mid \ldots)$ |
| LexCat | $P, c_P$ | $P, exp, c_P$ | $P, exp, H\#c_P$ | $S\#H, exp, P$ | $P=\mathrm{TOP}$ | – | – |
| LexCatDep | $P, c_P$ | $P, exp, c_P$ | $P, exp, H\#c_P$ | $S\#H, exp, P\#c_P$ | $P=\mathrm{TOP}$ | – | – |
| HeadWord | $P, c_P\#w_P$ | $P, exp, c_P\#w_P$ | $P, exp, H\#c_P\#w_P$ | $S\#H, exp, P$ | $P=\mathrm{TOP}$ | $c_S$ | $c_P$ |
| HWDep | $P, c_P\#w_P$ | $P, exp, c_P\#w_P$ | $P, exp, H\#c_P\#w_P$ | $S\#H, exp, P$ | $P=\mathrm{TOP}$ | $c_S\#P, H, S, w_P$ | $c_P$ |
| HWDep$\Delta$ | $P, c_P\#\Delta^{L,R}{}_P\#w_P$ | $P, exp, c_P\#\Delta^{L,R}{}_P\#w_P$ | $P, exp, H\#\Delta^{L,R}{}_P\#c_P\#w_P$ | $S\#H, exp, P$ | $P=\mathrm{TOP}$ | $c_S\#P, H, S, w_P$ | $c_P$ |
| HWDepConj | $P, c_P, conj_P\#w_P$ | $P, exp, c_P, conj_P\#w_P$ | $P, exp, H, conj_P\#c_P\#w_P$ | $S\#H, exp, P$ | $P=\mathrm{TOP}$ | $c_S\#P, H, S, w_P$ | $c_P$ |

Table 2: The lexicalized models

| Model | NoParse | LexCat | LP | LR | BP | BR | $\langle P,H,S\rangle$ | $\langle S\rangle$ | $\langle\rangle$ | CM on $\langle\rangle$ | $\leq$2 CD |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Baseline | 6 | 87.7 | 72.8 | 72.4 | 78,3 | 77.9 | 75.7 | 81.1 | 84.3 | 23.0 | 51.1 |
| *Conj* | 9 | 87.8 | 73.8 | 73.9 | 79.3 | 79.3 | 76.7 | 82.0 | 85.1 | 24.3 | 53.2 |
| Grandparent | 91 | 88.8 | 77.1 | 77.6 | 82.4 | 82.9 | 79.9 | 84.7 | 87.9 | 30.9 | 63.8 |
| $\Delta_{\mathrm{Pct}}$ | 6 | 88.1 | 73.7 | 73.1 | 79.2 | 78.6 | 76.5 | 81.8 | 84.9 | 23.1 | 53.2 |
| $\Delta_{\mathrm{Verb}}$ | 6 | 88.0 | 75.9 | 75.5 | 81.6 | 81.1 | 76.9 | 82.3 | 85.3 | 25.2 | 55.1 |
| $\Delta_{\mathrm{Adjacency}}$ | 6 | 88.6 | 77.5 | 77.3 | 82.9 | 82.8 | 78.9 | 83.8 | 86.9 | 24.8 | 59.6 |
| LexCat | 9 | 88.5 | 75.8 | 76.0 | 81.3 | 81.5 | 78.6 | 83.7 | 86.8 | 27.4 | 57.8 |
| LexCatDep | 9 | 88.5 | 75.7 | 75.9 | 81.2 | 81.4 | 78.4 | 83.5 | 86.6 | 26.3 | 57.9 |
| HeadWord | 8 | 89.6 | 77.9 | 78.0 | 83.0 | 83.1 | 80.5 | 85.2 | 88.3 | 30.4 | 63.0 |
| HWDep | 8 | 92.0 | 81.6 | 81.9 | 85.5 | 85.9 | 84.0 | 87.8 | 90.1 | 37.9 | 69.2 |
| HWDep$\Delta$ | 8 | 90.9 | 81.4 | 81.6 | 86.1 | 86.3 | 83.0 | 87.0 | 89.8 | 35.7 | 68.7 |
| HWDepConj | 9 | 91.8 | 80.7 | 81.2 | 84.8 | 85.3 | 83.6 | 87.5 | 89.9 | 36.5 | 68.6 |
| HWDep (+ tagger) | 7 | 91.7 | 81.4 | 81.8 | 85.6 | 85.9 | 83.6 | 87.5 | 89.9 | 38.1 | 69.1 |

Table 3: Performance of the models: LexCat indicates accuracy of the lexical categories; LP, LR, BP and BR (the standard Parseval scores labeled/bracketed precision and recall) are not commensurate with other Treebank parsers. $\langle P,H,S\rangle$, $\langle S\rangle$, and $\langle\rangle$ are as defined in section 2. CM on $\langle\rangle$ is the percentage of sentences with complete match on $\langle\rangle$, and $\leq$2 CD is the percentage of sentences with under 2 "crossing dependencies" as defined by $\langle\rangle$.

The $\langle P,H,S\rangle$ labeled dependencies we report are not directly comparable with Collins (1999), since CCG categories encode subcategorization frames. For instance, if the direct object of a verb has been recognized as such, but a PP has been mistaken as a complement (whereas the gold standard says it is an adjunct), the fully labeled dependency evaluation $\langle P,H,S\rangle$ will not award a point. Therefore, we also include in Table 3 a more comparable evaluation $\langle S\rangle$ which only takes the correctness of the non-head category into account. The reported figures are also deflated by retaining verb features like tensed/untensed. If this is done (by stripping off all verb features), an improvement of 0.6% on the $\langle P,H,S\rangle$ score for our best model is obtained.

### 5.3 Combining lexical and non-lexical information

When incorporating the adjacency distance measure or the coordination feature into the dependency model (**HWDep$\Delta$** and **HWDepConj**), overall performance is lower than with the dependency model alone. We conjecture that this arises from data sparseness. It cannot be concluded from these results alone that the lexical dependencies make structural information redundant or superfluous. Instead, it is quite likely that we are facing an estimation problem similar to Charniak (1999), who reports that the inclusion of the grandparent feature worsens performance of an MLE model, but improves performance if the individual distributions are modelled using Maximum Entropy. This intuition is strengthened by the fact that, on casual inspection of the scores for individual sentences, it is sometimes the case that the lexicalized models perform worse than the unlexicalized models.

### 5.4 The impact of tagging errors

All of the experiments described above use the POS-tags as given by CCGbank (which are the Treebank tags, with some corrections necessary to acquire correct features on categories). It is reasonable to assume that this input is of higher quality than can be produced by a POS-tagger. We therefore ran the dependency model on a test corpus tagged with the POS-tagger of Ratnaparkhi (1996), which is trained on the original Penn Treebank (see **HWDep (+ tagger)** in Table 3). Performance degrades slightly, which is to be expected, since our approach makes so much use of the POS-tag information for unknown words. However, a POS-tagger trained on CCGbank might yield slightly better results.

### 5.5 Limitations of the current model

Unlike Clark et al. (2002), our parser does not always model the dependencies in the logical form. For example, in the interpretation of a coordinate structure like *"buy and sell shares"*, *shares* will head an object of both *buy* and *sell*. Similarly, in examples like *"buy the company that wins"*, the relative construction makes *company* depend upon both *buy* as object and *wins* as subject. As is well known (Abney, 1997), DAG-like dependencies cannot in general be modeled with a generative approach of the kind taken here[3].

### 5.6 Comparison with Clark et al. (2002)

Clark et al. (2002) presents another statistical CCG parser, which is based on a conditional (rather than generative) model of the derived dependency structure, including non-surface dependencies. The following table compares the two parsers according to the evaluation of surface and deep dependencies given in Clark et al. (2002). We use Clark et al.'s parser to generate these dependencies from the output of our parser (see Clark and Hockenmaier (2002)) [4].

| | LP | LR | UP | UR |
|---|---|---|---|---|
| Clark | 81.9% | 81.8% | 89.1% | 90.1% |
| Hockenmaier | 83.7% | 84.2% | 90.5% | 91.1% |

## 6 Performance on specific constructions

One of the advantages of CCG is that it provides a simple, surface grammatical analysis of extraction and coordination. We investigate whether our best model, **HWDep**, predicts the correct analyses, using the development section 00.

**Coordination** There are two instances of argument cluster coordination (constructions like *cost \$5,000 in July and \$6,000 in August*) in the development corpus. Of these, **HWDep** recovers none correctly. This is a shortcoming in the model, rather than in CCG: the relatively high probability both of the NP modifier analysis of PPs like *in July* and of NP coordination is enough to misdirect the parser.

There are 203 instances of verb phrase coordination ($\mathsf{S[.]\backslash NP}$, with [.] any verbal feature) in the development corpus. On these, we obtain a labeled recall and precision of 67.0%/67.3%. Interestingly, on the 24 instances of right node raising (coordination of $\mathsf{(S[.]\backslash NP)/NP}$), our parser achieves higher performance, with labeled recall and precision of 79.2% and 73.1%. Figure 2 gives an example of the output of our parser on such a sentence.

**Extraction** Long-range dependencies are not captured by the evaluation used here. However, the accuracy for recovering lexical categories for words with "extraction" categories, such as relative pronouns, gives some indication of how well the model detects the presence of such dependencies.

The most common category for subject relative pronouns, $\mathsf{(NP\backslash NP)/(S[dcl]\backslash NP)}$, has been recovered with precision and recall of 97.1% (232 out of 239) and 94.3% (232/246).

Embedded subject extraction requires the special lexical category $\mathsf{((S[dcl]\backslash NP)/NP)/(S[dcl]\backslash NP)}$ for verbs like *think*. On this category, the model achieves a precision of 100% (5/5) and recall of 83.3% (5/6). The case the parser misanalyzed is due to lexical coverage: the verb *agree* occurs in our lexicon, but not with this category.

The most common category for object relative pronouns, $\mathsf{(NP\backslash NP)/(S[dcl]/NP)}$, has a recall of 76.2% (16 out of 21) and precision of 84.2% (16/19).

Free object relatives, $\mathsf{NP/(S[dcl]/NP)}$, have a recall of 84.6% (11/13), and precision of 91.7% (11/12). However, object extraction appears more frequently as a reduced relative (*the man John saw*), and there are no lexical categories indicating this extraction. Reduced relative clauses are captured by a type-changing rule $\mathsf{NP\backslash NP} \rightarrow \mathsf{S[dcl]/NP}$. This rule was applied 56 times in the gold standard, and 70

[3]It remains to be seen whether the more restricted reentrancies of CCG will ultimately support a generative model.

[4]Due to the smaller grammar and lexicon of Clark et al., our parser can only be evaluated on slightly over 94% of the sentences in section 23, whereas the figures for Clark et al. (2002) are on 97%.

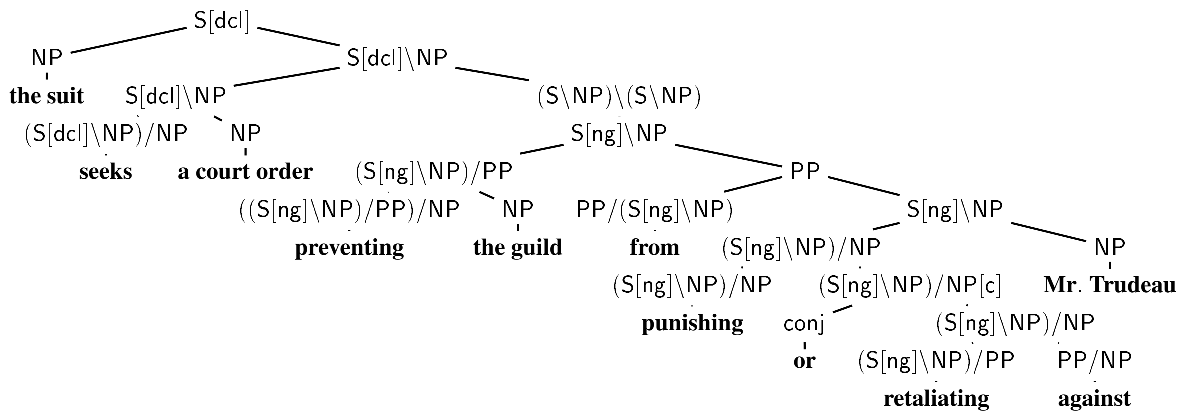



Figure 2: Right node raising output produced by our parser. *Punishing* and *retaliating* are unknown words.

times by the parser, out of which 48 times it corresponded to a rule in the gold standard (or 34 times, if the exact bracketing of the S[dcl]/NP is taken into account—this lower figure is due to attachment decisions made elsewhere in the tree).

These figures are difficult to compare with standard Treebank parsers. Despite the fact that the original Treebank does contain traces for movement, none of the existing parsers try to generate these traces (with the exception of Collins' Model 3, for which he only gives an overall score of 96.3%/98.8% P/R for subject extraction and 81.4%/59.4% P/R for other cases). The only "long range" dependency for which Collins gives numbers is subject extraction ⟨*SBAR, WHNP, SG, R*⟩, which has labeled precision and recall of 90.56% and 90.56%, whereas the CCG model achieves a labeled precision and recall of 94.3% and 96.5% on the most frequency subject extraction dependency ⟨NP\NP, (NP\NP)/(S[dcl]\NP), S[dcl]\NP⟩, which occurs 262 times in the gold standard and was produced 256 times by our parser. However, out of the 15 cases of this relation in the gold standard that our parser did not return, 8 were in fact analyzed as subject extraction of bare infinitivals ⟨NP\NP, (NP\NP)/(S[b]\NP), S[b]\NP⟩, yielding a combined recall of 97.3%.

## 7 Lexical coverage

The most serious problem facing parsers like the present one with large category sets is not so much the standard problem of unseen words, but rather the problem of words that have been seen, but not with the necessary category.

For standard Treebank parsers, the latter problem does not have much impact, if any, since the Penn Treebank tagset is fairly small, and the grammar underlying the Treebank is very permissive. However, for CCG this is a serious problem: the first three rows in table 4 show a significant difference in performance for sentences with complete lexical coverage ("No missing") and sentences with missing lexical entries ("Missing").

Using the POS-tags in the corpus, we can estimate the lexical probabilities $P(w \mid c)$ using a linear interpolation between the relative frequency estimates $\hat{P}(w \mid c)$ and the following approximation:[5]

$$\tilde{P}_{tags}(w \mid c) \quad = \quad \sum_{t \in tags} \hat{P}(w \mid t)\hat{P}(t \mid c)$$

We smooth the lexical probabilities as follows:

$$\tilde{P}(w \mid c) \quad = \quad \lambda\hat{P}(w \mid c) + (1-\lambda)\tilde{P}_{tags}(w \mid c)$$

Table 4 shows the performance of the baseline model with a frequency cutoff of 5 and 10 for rare words and with a smoothed and non-smoothed lexicon.[6] This frequency cutoff plays an important role here - smoothing with a small cutoff yields worse performance than not smoothing, whereas smoothing with a cutoff of 10 does not have a significant impact on performance. Smoothing the lexicon in this way does make the parser more robust, resulting in complete coverage of the test set. However, it does not affect overall performance, nor does it alleviate the problem for sentences with missing lexical entries for seen words.

[5]We compute $\lambda$ in the same way as Collins (1999), p. 185.

[6]Smoothing was only done for categories with a total frequency of 100 or more.

| | Baseline, Cutoff = 5 (Missing = 463 sentences) | | Baseline, Cutoff = 10 (Missing = 387 sentences) | | HWDep, Cutoff = 10 (Missing = 387 sentences) |
|---|---|---|---|---|---|
| | Non-smoothed | Smoothed | Non-smoothed | Smoothed | Smoothed |
| Parse failures | 6 | – | 5 | – | – |
| $\langle P, H, S \rangle$, All | 75.7 | 73.2 | 76.2 | 76.3 | 83.9 |
| $\langle P, H, S \rangle$, Missing | 66.4 | 64.2 | 67.0 | 67.1 | 75.1 |
| $\langle P, H, S \rangle$, No missing | 78.5 | 75.9 | 78.5 | 78.6 | 86.6 |

Table 4: The impact of lexical coverage, using a different cutoff for rare words and smoothing (section 23)

## 8 Conclusion and future work

We have compared a number of generative probability models of CCG derivations, and shown that our best model recovers 89.9% of word-word dependencies on section 23 of CCGbank. On section 00, it recovers 89.7% of word-word dependencies. These figures are surprisingly close to the figure of 90.9% reported by Collins (1999) on section 00, given that, in order to allow a direct comparison, we have used the same interpolation technique and beam strategy as Collins (1999), which are very unlikely to be as well-tuned to our kind of grammar.

As is to be expected, a statistical model of a CCG extracted from the Treebank is less robust than a model with an overly permissive grammar such as Collins (1999). This problem seems to stem mainly from the incomplete coverage of the lexicon. We have shown that smoothing can compensate for entirely unknown words. However, this approach does not help on sentences which require previously unseen entries for known words. We would expect a less naive approach such as applying morphological rules to the observed entries, together with better smoothing techniques, to yield better results.

We have also shown that a statistical model of CCG benefits from word-word dependencies to a much greater extent than a less linguistically motivated model such as Collins' Model 1. This indicates to us that, although the task faced by a CCG parser might seem harder prima facie, there are advantages to using a more linguistically adequate grammar.

## Acknowledgements

Thanks to Stephen Clark, Miles Osborne and the ACL-02 referees for comments. Various parts of the research were funded by EPSRC grants GR/M96889 and GR/R02450 and an EPSRC studentship.

## References

Steven Abney. 1997. Stochastic Attribute-Value Grammars. *Computational Linguistics*, 23(4).

Bob Carpenter. 1992. Categorial Grammars, Lexical Rules, and the English Predicative. In R. Levine, ed., *Formal Grammar: Theory and Implementation*. OUP.

Eugene Charniak. 1999. A Maximum-Entropy-Inspired Parser. TR CS-99-12, Brown University.

David Chiang. 2000. Statistical Parsing with an Automatically-Extracted Tree Adjoining Grammar *38th ACL,* Hong Kong, pp. 456-463.

Stephen Clark and Julia Hockenmaier. 2002. Evaluating a Wide-Coverage CCG Parser. *LREC Beyond PARSEVAL workshop*, Las Palmas, Spain.

Stephen Clark, Julia Hockenmaier, and Mark Steedman. 2002. Building Deep Dependency Structures Using a Wide-Coverage CCG Parser. *40th ACL,* Philadelphia.

Michael Collins. 1997. Three Generative Lexicalized Models for Statistical Parsing. *35th ACL*, Madrid, pp. 16–23.

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

Daniel Gildea. 2001. Corpus Variation and Parser Performance. *EMNLP*, Pittsburgh, PA.

Julia Hockenmaier. 2001. Statistical Parsing for CCG with Simple Generative Models. *Student Workshop, 39th ACL/ 10th EACL*, Toulouse, France, pp. 7–12.

Julia Hockenmaier and Mark Steedman 2002. Acquiring Compact Lexicalized Grammars from a Cleaner Treebank. *Third LREC*, Las Palmas, Spain.

Joshua Goodman. 1997. Probabilistic Feature Grammars. *IWPT*, Boston.

Mark Johnson. 1998. PCFG Models of Linguistic Tree Representations. *Computational Linguistics*, 24(4).

Adwait Ratnaparkhi. 1996. A Maximum Entropy Part-Of-Speech Tagger. *EMNLP*, Philadelphia, pp. 133–142.

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