In Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics, 2002, Philadelphia.

Parsing the Wall Street Journal using a Lexical-Functional Grammar and Discriminative Estimation Techniques

Stefan Riezler
Palo Alto Research Center
Palo Alto, CA 94304
riezler@parc.com

Tracy H. King
Palo Alto Research Center
Palo Alto, CA 94304
thking@parc.com

Ronald M. Kaplan
Palo Alto Research Center
Palo Alto, CA 94304
kaplan@parc.com

Richard Crouch
Palo Alto Research Center
Palo Alto, CA 94304
crouch@parc.com

John T. Maxwell III
Palo Alto Research Center
Palo Alto, CA 94304
maxwell@parc.com

Mark Johnson
Brown University
Providence, RI 02912
mj@cs.brown.edu

Abstract

We present a stochastic parsing system consisting of a Lexical-Functional Grammar (LFG), a constraint-based parser and a stochastic disambiguation model. We report on the results of applying this system to parsing the UPenn Wall Street Journal (WSJ) treebank. The model combines full and partial parsing techniques to reach full grammar coverage on unseen data. The treebank annotations are used to provide partially labeled data for discriminative statistical estimation using exponential models. Disambiguation performance is evaluated by measuring matches of predicate-argument relations on two distinct test sets. On a gold standard of manually annotated f-structures for a subset of the WSJ treebank, this evaluation reaches 79% F-score. An evaluation on a gold standard of dependency relations for Brown corpus data achieves 76% F-score.

1 Introduction

Statistical parsing using combined systems of hand-coded linguistically fine-grained grammars and stochastic disambiguation components has seen considerable progress in recent years. However, such attempts have so far been confined to a relatively small scale for various reasons. Firstly, the rudimentary character of functional annotations in standard treebanks has hindered the direct use of such data for statistical estimation of linguistically fine-grained statistical parsing systems. Rather, parameter estimation for such models had to resort to unsupervised techniques (Bouma et al., 2000; Riezler et al., 2000), or training corpora tailored to the specific grammars had to be created by parsing and manual disambiguation, resulting in relatively small training sets of around 1,000 sentences (Johnson et al., 1999). Furthermore, the effort involved in coding broad-coverage grammars by hand has often led to the specialization of grammars to relatively small domains, thus sacrificing grammar coverage (i.e. the percentage of sentences for which at least one analysis is found) on free text. The approach presented in this paper is a first attempt to scale up stochastic parsing systems based on linguistically fine-grained hand-coded grammars to the UPenn Wall Street Journal (henceforth WSJ) treebank (Marcus et al., 1994).

The problem of grammar coverage, i.e. the fact that not all sentences receive an analysis, is tackled in our approach by an extension of a full-fledged Lexical-Functional Grammar (LFG) and a constraint-based parser with partial parsing techniques. In the absence of a complete parse, a so-called "FRAGMENT grammar" allows the input to be analyzed as a sequence of well-formed chunks. The set of fragment parses is then chosen on the basis of a fewest-chunk method. With this combination of full and partial parsing techniques we achieve 100% grammar coverage on unseen data.

Another goal of this work is the best possible exploitation of the WSJ treebank for discriminative estimation of an exponential model on LFG parses. We define discriminative or conditional criteria with re-

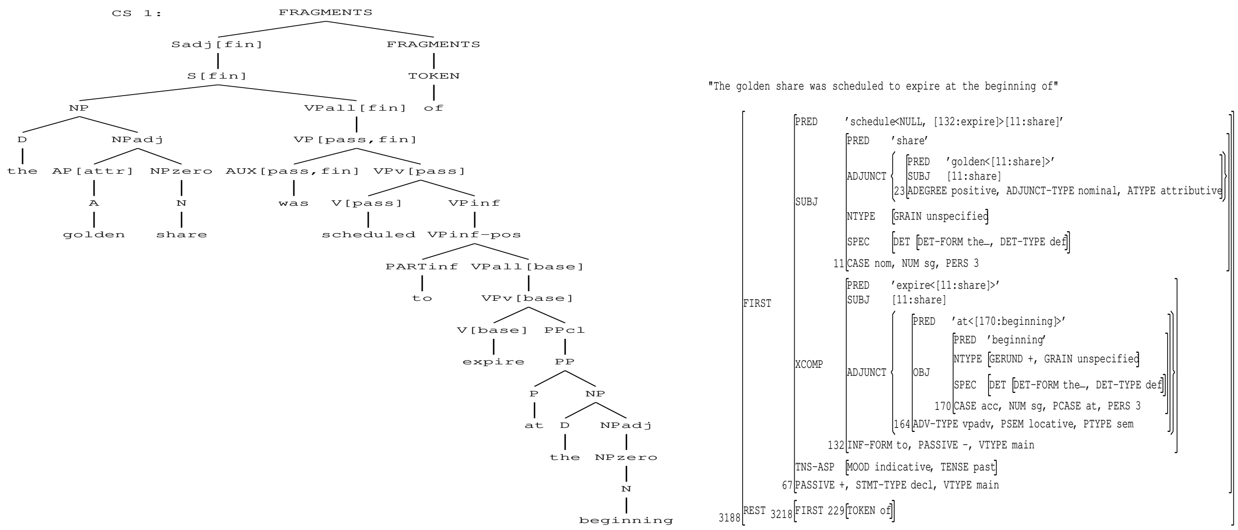

Figure 1: FRAGMENT c-/f-structure for *The golden share was scheduled to expire at the beginning of*

spect to the set of grammar parses consistent with the treebank annotations. Such data can be gathered by applying labels and brackets taken from the treebank annotation to the parser input. The rudimentary treebank annotations are thus used to provide partially labeled data for discriminative estimation of a probability model on linguistically fine-grained parses.

Concerning empirical evaluation of disambiguation performance, we feel that an evaluation measuring matches of predicate-argument relations is more appropriate for assessing the quality of our LFG-based system than the standard measure of matching labeled bracketing on section 23 of the WSJ treebank. The first evaluation we present measures matches of predicate-argument relations in LFG f-structures (henceforth the LFG annotation scheme) to a gold standard of manually annotated f-structures for a representative subset of the WSJ treebank. The evaluation measure counts the number of predicate-argument relations in the f-structure of the parse selected by the stochastic model that match those in the gold standard annotation. Our parser plus stochastic disambiguator achieves 79% F-score under this evaluation regime.

Furthermore, we employ another metric which maps predicate-argument relations in LFG f-structures to the dependency relations (henceforth the DR annotation scheme) proposed by Carroll et al. (1999). Evaluation with this metric measures the matches of dependency relations to Carroll et al.'s gold standard corpus. For a direct comparison of our results with Carroll et al.'s system, we computed an F-score that does not distinguish different types of dependency relations. Under this measure we obtain 76% F-score.

This paper is organized as follows. Section 2 describes the Lexical-Functional Grammar, the constraint-based parser, and the robustness techniques employed in this work. In section 3 we present the details of the exponential model on LFG parses and the discriminative statistical estimation technique. Experimental results are reported in section 4. A discussion of results is in section 5.

2 Robust Parsing using LFG

2.1 A Broad-Coverage LFG

The grammar used for this project was developed in the ParGram project (Butt et al., 1999). It uses LFG as a formalism, producing c(onstituent)-structures (trees) and f(unctional)-structures (attribute value matrices) as output. The c-structures encode constituency. F-structures encode predicate-argument relations and other grammatical information, e.g., number, tense. The XLE parser (Maxwell and Kaplan, 1993) was used to produce packed representations, specifying all possible grammar analyses of the input.

The grammar has 314 rules with regular expression right-hand sides which compile into a collection of finite-state machines with a total of 8,759 states and 19,695 arcs. The grammar uses several lexicons and two guessers: one guesser for words recognized by the morphological analyzer but not in the lexicons and one for those not recognized. As such, most nouns, adjectives, and adverbs have no explicit lexical entry. The main verb lexicon contains 9,652 verb stems and 23,525 subcategorization frame-verb stem entries; there are also lexicons for adjectives and nouns with subcategorization frames and for closed class items.

For estimation purposes using the WSJ treebank, the grammar was modified to parse part of speech tags and labeled bracketing. A stripped down version of the WSJ treebank was created that used only those POS tags and labeled brackets relevant for determining grammatical relations. The WSJ labeled brackets are given LFG lexical entries which constrain both the c-structure and the f-structure of the parse. For example, the WSJ's ADJP-PRD label must correspond to an AP in the c-structure and an XCOMP in the f-structure. In this version of the corpus, all WSJ labels with -SBJ are retained and are restricted to phrases corresponding to SUBJ in the LFG grammar; in addition, it contains NP under VP (OBJ and OBJth in the LFG grammar), all -LGS tags (OBL-AG), all -PRD tags (XCOMP), VP under VP (XCOMP), SBAR- (COMP), and verb POS tags under VP (V in the c-structure). For example, our labeled bracketing of wsj_1305.mrg is *[NP-SBJ His credibility] is/VBZ_ also [PP-PRD on the line] in the investment community.*

Some mismatches between the WSJ labeled bracketing and the LFG grammar remain. These often arise when a given constituent fills a grammatical role in more than one clause. For example, in wsj_1303.mrg *Japan's Daiwa Securities Co. named Masahiro Dozen president.*, the noun phrase *Masahiro Dozen* is labeled as an NP-SBJ. However, the LFG grammar treats it as the OBJ of the matrix clause. As a result, the labeled bracketed version of this sentence does not receive a full parse, even though its unlabeled, string-only counterpart is well-formed. Some other bracketing mismatches remain, usually the result of adjunct attachment. Such mismatches occur in part because, besides minor modifications to match the bracketing for special constructions, e.g., negated infinitives, the grammar was not altered to mirror the idiosyncrasies of the WSJ bracketing.

2.2 Robustness Techniques

To increase robustness, the standard grammar has been augmented with a FRAGMENT grammar. This grammar parses the sentence as well-formed chunks specified by the grammar, in particular as Ss, NPs, PPs, and VPs. These chunks have both c-structures and f-structures corresponding to them. Any token that cannot be parsed as one of these chunks is parsed as a TOKEN chunk. The TOKENs are also recorded in the c- and f-structures. The grammar has a fewest-chunk method for determining the correct parse. For example, if a string can be parsed as two NPs and a VP or as one NP and an S, the NP-S option is chosen. A sample FRAGMENT c-structure and f-structure are shown in Fig. 1 for wsj_0231.mrg (*The golden share was scheduled to expire at the beginning of*), an incomplete sentence; the parser builds one S chunk and then one TOKEN for the stranded preposition.

A final capability of XLE that increases coverage of the standard-plus-fragment grammar is a SKIMMING technique. Skimming is used to avoid timeouts and memory problems. When the amount of time or memory spent on a sentence exceeds a threshhold, XLE goes into skimming mode for the constituents whose processing has not been completed. When XLE skims these remaining constituents, it does a bounded amount of work per subtree. This guarantees that XLE finishes processing a sentence in a polynomial amount of time. In parsing section 23, 7.2% of the sentences were skimmed; 26.1% of these resulted in full parses, while 73.9% were FRAGMENT parses.

The grammar coverage achieved 100% of section 23 as unseen unlabeled data: 74.7% as full parses, 25.3% FRAGMENT and/or SKIMMED parses.

3 Discriminative Statistical Estimation from Partially Labeled Data

3.1 Exponential Models on LFG Parses

We employed the well-known family of exponential models for stochastic disambiguation. In this paper we are concerned with conditional exponential models of the form:

$$p_{\boldsymbol{\lambda}}(x|y) = Z_{\boldsymbol{\lambda}}(y)^{-1} e^{\boldsymbol{\lambda} \cdot \boldsymbol{f}(x)}$$

where $X(y)$ is the set of parses for sentence y, $Z_{\boldsymbol{\lambda}}(y) = \sum_{x \in X(y)} e^{\boldsymbol{\lambda} \cdot \boldsymbol{f}(x)}$ is a normalizing constant, $\boldsymbol{\lambda} = (\lambda_1, \ldots, \lambda_n) \in \mathbb{R}^n$ is a vector of log-parameters, $\boldsymbol{f} = (f_1, \ldots, f_n)$ is a vector of property-functions $f_i : \mathcal{X} \to \mathbb{R}$ for $i = 1, \ldots, n$ on the set of parses $\mathcal{X}$, and $\boldsymbol{\lambda} \cdot \boldsymbol{f}(x)$ is the vector dot product $\sum_{i=1}^{n} \lambda_i f_i(x)$.

In our experiments, we used around 1000 complex property-functions comprising information about c-structure, f-structure, and lexical elements in parses, similar to the properties used in Johnson et al. (1999). For example, there are property functions for c-structure nodes and c-structure subtrees, indicating attachment preferences. High versus low attachment is indicated by property functions counting the number of recursively embedded phrases. Other property functions are designed to refer to f-structure attributes, which correspond to grammatical functions in LFG, or to atomic attribute-value pairs in f-structures. More complex property functions are designed to indicate, for example, the branching behaviour of c-structures and the (non)-parallelism of coordinations on both c-structure and f-structure levels. Furthermore, properties refering to lexical elements based on an auxiliary distribution approach as presented in Riezler et al. (2000) are included in the model. Here tuples of head words, argument words, and grammatical relations are extracted from the training sections of the WSJ, and fed into a finite mixture model for clustering grammatical relations. The clustering model itself is then used to yield smoothed probabilities as values for property functions on head-argument-relation tuples of LFG parses.

3.2 Discriminative Estimation

Discriminative estimation techniques have recently received great attention in the statistical machine learning community and have already been applied to statistical parsing (Johnson et al., 1999; Collins, 2000; Collins and Duffy, 2001). In discriminative estimation, only the conditional relation of an analysis given an example is considered relevant, whereas in maximum likelihood estimation the joint probability of the training data to best describe observations is maximized. Since the discriminative task is kept in mind during estimation, discriminative methods can yield improved performance. In our case, discriminative criteria cannot be defined directly with respect to "correct labels" or "gold standard" parses since the WSJ annotations are not sufficient to disambiguate the more complex LFG parses. However, instead of retreating to unsupervised estimation techniques or creating small LFG treebanks by hand, we use the labeled bracketing of the WSJ training sections to guide discriminative estimation. That is, discriminative criteria are defined with respect to the *set of parses consistent with the WSJ annotations*.[1]

The objective function in our approach, denoted by $P(\boldsymbol{\lambda})$, is the joint of the negative log-likelihood $-L(\boldsymbol{\lambda})$ and a Gaussian regularization term $-G(\boldsymbol{\lambda})$ on the parameters $\boldsymbol{\lambda}$. Let $\{(y_j, z_j)\}_{j=1}^{m}$ be a set of training data, consisting of pairs of sentences y and partial annotations z, let $X(y, z)$ be the set of parses for sentence y consistent with annotation z, and let $X(y)$ be the set of all parses produced by the grammar for sentence y. Furthermore, let $p[f]$ denote the expectation of function f under distribution p. Then $P(\boldsymbol{\lambda})$ can be defined for a conditional exponential model $p_{\boldsymbol{\lambda}}(z|y)$ as:

$$\begin{aligned}
P(\boldsymbol{\lambda}) &= -L(\boldsymbol{\lambda}) - G(\boldsymbol{\lambda}) \\
&= -\log \prod_{j=1}^{m} p_{\boldsymbol{\lambda}}(z_j|y_j) + \sum_{i=1}^{n} \frac{\lambda_i^2}{2\sigma_i^2} \\
&= -\sum_{j=1}^{m} \log \frac{\sum_{X(y_j,z_j)} e^{\boldsymbol{\lambda} \cdot \boldsymbol{f}(x)}}{\sum_{X(y_j)} e^{\boldsymbol{\lambda} \cdot \boldsymbol{f}(x)}} + \sum_{i=1}^{n} \frac{\lambda_i^2}{2\sigma_i^2} \\
&= -\sum_{j=1}^{m} \log \sum_{X(y_j,z_j)} e^{\boldsymbol{\lambda} \cdot \boldsymbol{f}(x)} \\
&\quad + \sum_{j=1}^{m} \log \sum_{X(y_j)} e^{\boldsymbol{\lambda} \cdot \boldsymbol{f}(x)} + \sum_{i=1}^{n} \frac{\lambda_i^2}{2\sigma_i^2}.
\end{aligned}$$

Intuitively, the goal of estimation is to find model pa-

[1] An earlier approach using partially labeled data for estimating stochastics parsers is Pereira and Schabes's (1992) work on training PCFG from partially bracketed data. Their approach differs from the one we use here in that Pereira and Schabes take an EM-based approach maximizing the joint likelihood of the parses and strings of their training data, while we maximize the conditional likelihood of the sets of parses given the corresponding strings in a discriminative estimation setting.

rameters which make the two expectations in the last equation equal, i.e. which adjust the model parameters to put all the weight on the parses consistent with the annotations, modulo a penalty term from the Gaussian prior for too large or too small weights.

Since a closed form solution for such parameters is not available, numerical optimization methods have to be used. In our experiments, we applied a conjugate gradient routine, yielding a fast converging optimization algorithm where at each iteration the negative log-likelihood $P(\boldsymbol{\lambda})$ and the gradient vector have to be evaluated.[2] For our task the gradient takes the form:

$$\nabla P(\boldsymbol{\lambda}) = \left\langle \frac{\partial P(\boldsymbol{\lambda})}{\partial \lambda_1}, \frac{\partial P(\boldsymbol{\lambda})}{\partial \lambda_2}, \ldots, \frac{\partial P(\boldsymbol{\lambda})}{\partial \lambda_n} \right\rangle, \text{ and}$$

$$\begin{aligned} \frac{\partial P(\boldsymbol{\lambda})}{\partial \lambda_i} &= -\sum_{j=1}^{m} \Big(\sum_{x \in X(y_j, z_j)} \frac{e^{\boldsymbol{\lambda} \cdot \boldsymbol{f}(x)} f_i(x)}{\sum_{x \in X(y_j, z_j)} e^{\boldsymbol{\lambda} \cdot \boldsymbol{f}(x)}} \\ &\quad - \sum_{x \in X(y_j)} \frac{e^{\boldsymbol{\lambda} \cdot \boldsymbol{f}(x)} f_i(x)}{\sum_{x \in X(y_j)} e^{\boldsymbol{\lambda} \cdot \boldsymbol{f}(x)}} \Big) + \frac{\lambda_i}{\sigma_i^2}. \end{aligned}$$

The derivatives in the gradient vector intuitively are again just a difference of two expectations

$$-\sum_{j=1}^{m} p_{\boldsymbol{\lambda}}[f_i | y_j, z_j] + \sum_{j=1}^{m} p_{\boldsymbol{\lambda}}[f_i | y_j] + \frac{\lambda_i}{\sigma_i^2}.$$

Note also that this expression shares many common terms with the likelihood function, suggesting an efficient implementation of the optimization routine.

4 Experimental Evaluation

4.1 Training

The basic training data for our experiments are sections 02-21 of the WSJ treebank. As a first step, all sections were parsed, and the packed parse forests unpacked and stored. For discriminative estimation, this data set was restricted to sentences which receive a full parse (in contrast to a FRAGMENT or SKIMMED parse) for both its partially labeled and its unlabeled variant. Furthermore, only sentences which received at most 1,000 parses were used. From this set, sentences of which a discriminative learner cannot possibly take advantage, i.e. sentences where the set of parses assigned to the partially labeled string was not a proper subset of the parses assigned the unlabeled string, were removed. These successive selection steps resulted in a final training set consisting of 10,000 sentences, each with parses for partially labeled and unlabeled versions. Altogether there were 150,000 parses for partially labeled input and 500,000 for unlabeled input.

For estimation, a simple property selection procedure was applied to the full set of around 1000 properties. This procedure is based on a frequency cutoff on instantiations of properties for the parses in the labeled training set. The result of this procedure is a reduction of the property vector to about half its size. Furthermore, a held-out data set was created from section 24 of the WSJ treebank for experimental selection of the variance parameter of the prior distribution. This set consists of 120 sentences which received only full parses, out of which the most plausible one was selected manually.

4.2 Testing

Two different sets of test data were used: (i) 700 sentences randomly extracted from section 23 of the WSJ treebank and given gold-standard f-structure annotations according to our LFG scheme, and (ii) 500 sentences from the Brown corpus given gold standard annotations by Carroll et al. (1999) according to their dependency relations (DR) scheme.[3]

Annotating the WSJ test set was bootstrapped by parsing the test sentences using the LFG grammar and also checking for consistency with the Penn Treebank annotation. Starting from the (sometimes fragmentary) parser analyses and the Treebank annotations, gold standard parses were created by manual corrections and extensions of the LFG parses. Manual corrections were necessary in about half of the cases. The average sentence length of the WSJ f-structure bank is 19.8 words; the average number of predicate-argument relations in the gold-standard f-structures is 31.2.

Performance on the LFG-annotated WSJ test set

[2] An alternative numerical method would be a combination of iterative scaling techniques with a conditional EM algorithm (Jebara and Pentland, 1998). However, it has been shown experimentally that conjugate gradient techniques can outperform iterative scaling techniques by far in running time (Minka, 2001).

[3] Both corpora are available online. The WSJ f-structure bank at `www.parc.com/istl/groups/nltt/fsbank/`, and Carroll et al.'s corpus at `www.cogs.susx.ac.uk/lab/nlp/carroll/greval.html`.

was measured using both the LFG and DR metrics, thanks to an f-structure-to-DR annotation mapping. Performance on the DR-annotated Brown test set was only measured using the DR metric.

The LFG evaluation metric is based on the comparison of full f-structures, represented as triples $relation(predicate, argument)$. The predicate-argument relations of the f-structure for one parse of the sentence *Meridian will pay a premium of $30.5 million to assume $2 billion in deposits.* are shown in Fig. 2.

number($:9, billion:17)	number($:24, million:4)
detform(premium:3, a)	mood(pay:0, indicative)
tense(pay:0, fut)	adjunct(million:4, '30.5':28)
adjunct(premium:3, of:23)	adjunct(billion:17, '2':19)
adjunct($:9, in:11)	adjunct(pay:0, assume:7)
obj(pay:0, premium:3)	stmttype(pay:0, decl)
subj(pay:0, 'Meridian':5)	obj(assume:7, $:9)
obj(of:23, $:24)	subj(assume:7, pro:8)
obj(in:11, deposit:12)	prontype(pro:8, null)
stmttype(assume:7, purpose)	

Figure 2: LFG predicate-argument relation representation

The DR annotation for our example sentence, obtained via a mapping from f-structures to Carroll et al's annotation scheme, is shown in Fig. 3.

(aux _ pay will)	(subj pay Meridian _)
(detmod _ premium a)	(mod _ million 30.5)
(mod _ $ million)	(mod of premium $)
(dobj pay premium _)	(mod _ billion 2)
(mod _ $ billion)	(mod in $ deposit)
(dobj assume $ _)	(mod to pay assume)

Figure 3: Mapping to Carroll et al.'s dependency-relation representation

Superficially, the LFG and DR representations are very similar. One difference between the annotation schemes is that the LFG representation in general specifies more relation tuples than the DR representation. Also, multiple occurences of the same lexical item are indicated explicitly in the LFG representation but not in the DR representation. The main conceptual difference between the two annotation schemes is the fact that the DR scheme crucially refers to phrase-structure properties and word order as well as to grammatical relations in the definition of dependency relations, whereas the LFG scheme abstracts away from serialization and phrase-structure. Facts like this can make a correct mapping of LFG f-structures to DR relations problematic. Indeed, we believe that we still underestimate by a few points because of DR mapping difficulties. [4]

4.3 Results

In our evaluation, we report F-scores for both types of annotation, LFG and DR, and for three types of parse selection, (i) *lower bound*: random choice of a parse from the set of analyses (averaged over 10 runs), (ii) *upper bound*: selection of the parse with the best F-score according to the annotation scheme used, and (iii) *stochastic*: the parse selected by the stochastic disambiguator. The *error reduction* row lists the reduction in error rate relative to the upper and lower bounds obtained by the stochastic disambiguation model. F-score is defined as $2 \times precision \times recall/(precision + recall)$.

Table 1 gives results for 700 examples randomly selected from section 23 of the WSJ treebank, using both LFG and DR measures.

Table 1: Disambiguation results for 700 randomly selected examples from section 23 of the WSJ treebank using LFG and DR measures.

	LFG	DR
upper bound	84.1	80.7
stochastic	78.6	73.0
lower bound	75.5	68.8
error reduction	36	35

The effect of the quality of the parses on disambiguation performance can be illustrated by breaking down the F-scores according to whether the parser yields full parses, FRAGMENT, SKIMMED, or SKIMMED+FRAGMENT parses for the test sentences. The percentages of test examples which belong to the respective classes of quality are listed in the first row of Table 2. F-scores broken down according to classes of parse quality are recorded in the follow-

[4]See Carroll et al. (1999) for more detail on the DR annotation scheme, and see Crouch et al. (2002) for more detail on the differences between the DR and the LFG annotation schemes, as well as on the difficulties of the mapping from LFG f-structures to DR annotations.

ing rows. The first column shows F-scores for all parses in the test set, as in Table 1. The second column shows the best F-scores when restricting attention to examples which receive only full parses. The third column reports F-scores for examples which receive only non-full parses, i.e. FRAGMENT or SKIMMED parses or SKIMMED+FRAGMENT parses. Columns 4-6 break down non-full parses according to examples which receive only FRAGMENT, only SKIMMED, or only SKIMMED+FRAGMENT parses.

Results of the evaluation on Carroll et al.'s Brown test set are given in Table 3. Evaluation results for the DR measure applied to the Brown corpus test set broken down according to parse-quality are shown in Table 2.

In Table 3 we show the DR measure along with an evaluation measure which facilitates a direct comparison of our results to those of Carroll et al. (1999). Following Carroll et al. (1999), we count a dependency relation as correct if the gold standard has a relation with the same governor and dependent but perhaps with a different relation-type. This dependency-only (DO) measure thus does not reflect mismatches between arguments and modifiers in a small number of cases. Note that since for the evaluation on the Brown corpus, no heldout data were available to adjust the variance parameter of a Bayesian model, we used a plain maximum-likelihood model for disambiguation on this test set.

Table 3: Disambiguation results on 500 Brown corpus examples using DO measure and DR measures.

	DO	DR
Carroll et al. (1999)	75.1	-
upper bound	82.0	80.0
stochastic	76.1	74.0
lower bound	73.3	71.7
error reduction	32	33

5 Discussion

We have presented a first attempt at scaling up a stochastic parsing system combining a hand-coded linguistically fine-grained grammar and a stochastic disambiguation model to the WSJ treebank. Full grammar coverage is achieved by combining specialized constraint-based parsing techniques for LFG grammars with partial parsing techniques. Furthermore, a maximal exploitation of treebank annotations for estimating a distribution on fine-grained LFG parses is achieved by letting grammar analyses which are consistent with the WSJ labeled bracketing define a gold standard set for discriminative estimation. The combined system trained on WSJ data achieves full grammar coverage and disambiguation performance of 79% F-score on WSJ data, and 76% F-score on the Brown corpus test set.

While disambiguation performance of around 79% F-score on WSJ data seems promising, from one perspective it only offers a 3% absolute improvement over a lower bound random baseline. We think that the high lower bound measure highlights an important aspect of symbolic constraint-based grammars (in contrast to treebank grammars): the symbolic grammar already significantly restricts/disambiguates the range of possible analyses, giving the disambiguator a much narrower window in which to operate. As such, it is more appropriate to assess the disambiguator in terms of reduction in error rate (36% relative to the upper bound) than in terms of absolute F-score. Both the DR and LFG annotations broadly agree in their measure of error reduction.

The lower reduction in error rate relative to the upper bound for DR evaluation on the Brown corpus can be attributed to a corpus effect that has also been observed by Gildea (2001) for training and testing PCFGs on the WSJ and Brown corpora.[5]

Breaking down results according to parse quality shows that irrespective of evaluation measure and corpus, around 4% overall performance is lost due to non-full parses, i.e. FRAGMENT, or SKIMMED, or SKIMMED+FRAGMENT parses.

Due to the lack of standard evaluation measures and gold standards for predicate-argument matching, a comparison of our results to other stochastic parsing systems is difficult. To our knowledge, so far the only direct point of comparison is the parser of Carroll et al. (1999) which is also evaluated on Carroll et al.'s test corpus. They report an F-score

[5]Gildea reports a decrease from 86.1%/86.6% recall/precision on labeled bracketing to 80.3%/81% when going from training and testing on the WSJ to training on the WSJ and testing on the Brown corpus.

Table 2: LFG F-scores for the 700 WSJ test examples and DR F-scores for the 500 Brown test examples broken down according to parse quality.

WSJ-LFG	all	full	non-full	fragments	skimmed	skimmed+fragments
% of test set	100	74.7	25.3	20.4	1.4	3.4
upper bound	84.1	88.5	73.4	76.7	70.3	61.3
stochastic	78.6	82.5	69.0	72.4	66.6	56.2
lower bound	75.5	78.4	67.7	71.0	63.0	55.9
Brown-DR	all	full	non-full	fragments	skimmed	skimmed+fragments
% of test set	100	79.6	20.4	20.0	2.0	1.6
upper bound	80.0	84.5	65.4	65.4	56.0	53.5
stochastic	74.0	77.9	61.5	61.5	52.8	50.0
lower bound	71.1	74.8	59.2	59.1	51.2	48.9

of 75.1% for a DO evaluation that ignores predicate labels, counting only dependencies. Under this measure, our system achieves 76.1% F-score.

References

Gosse Bouma, Gertjan von Noord, and Robert Malouf. 2000. Alpino: Wide-coverage computational analysis of Dutch. In *Proceedings of Computational Linguistics in the Netherlands*, Amsterdam, Netherlands.

Miriam Butt, Tracy King, Maria-Eugenia Niño, and Frédérique Segond. 1999. *A Grammar Writer's Cookbook*. Number 95 in CSLI Lecture Notes. CSLI Publications, Stanford, CA.

John Carroll, Guido Minnen, and Ted Briscoe. 1999. Corpus annotation for parser evaluation. In *Proceedings of the EACL workshop on Linguistically Interpreted Corpora (LINC)*, Bergen, Norway.

Michael Collins and Nigel Duffy. 2001. Convolution kernels for natural language. In *Advances in Neural Information Processing Systems 14(NIPS'01)*, Vancouver.

Michael Collins. 2000. Discriminative reranking for natural language processing. In *Proceedings of the Seventeenth International Conference on Machine Learning (ICML'00)*, Stanford, CA.

Richard Crouch, Ronald M. Kaplan, Tracy H. King, and Stefan Riezler. 2002. A comparison of evaluation metrics for a broad-coverage stochastic parser. In *Proceedings of the "Beyond PARSEVAL" Workshop at the 3rd International Conference on Language Resources and Evaluation (LREC'02)*, Las Palmas, Spain.

Dan Gildea. 2001. Corpus variation and parser performance. In *Proceedings of 2001 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Pittsburgh, PA.

Tony Jebara and Alex Pentland. 1998. Maximum conditional likelihood via bound maximization and the CEM algorithm. In *Advances in Neural Information Processing Systems 11 (NIPS'98)*.

Mark Johnson, Stuart Geman, Stephen Canon, Zhiyi Chi, and Stefan Riezler. 1999. Estimators for stochastic "unification-based" grammars. In *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics (ACL'99)*, College Park, MD.

Mitchell Marcus, Grace Kim, Mary Ann Marcinkiewicz, Robert MacIntyre, Ann Bies, Mark Ferguson, Karen Katz, and Britta Schasberger. 1994. The Penn treebank: Annotating predicate argument structure. In *ARPA Human Language Technology Workshop*.

John Maxwell and Ron Kaplan. 1993. The interface between phrasal and functional constraints. *Computational Linguistics*, 19(4):571–589.

Thomas Minka. 2001. Algorithms for maximum-likelihood logistic regression. Department of Statistics, Carnegie Mellon University.

Fernando Pereira and Yves Schabes. 1992. Inside-outside reestimation from partially bracketed corpora. In *Proceedings of the 30th Annual Meeting of the Association for Computational Linguistics (ACL'92)*, Newark, Delaware.

Stefan Riezler, Detlef Prescher, Jonas Kuhn, and Mark Johnson. 2000. Lexicalized Stochastic Modeling of Constraint-Based Grammars using Log-Linear Measures and EM Training. In *Proceedings of the 38th Annual Meeting of the Association for Computational Linguistics (ACL'00)*, Hong Kong.