A Comparison of Evaluation Metrics for a Broad-Coverage Stochastic Parser

Richard Crouch, Ronald M. Kaplan, Tracy H. King, Stefan Riezler

Palo Alto Research Center
3333 Coyote Hill Road
Palo Alto, CA, 94025
{crouch|kaplan|thking|riezler}@parc.com

Abstract

This paper reports on the use of two distinct evaluation metrics for assessing a stochastic parsing model consisting of a broad-coverage Lexical-Functional Grammar (LFG), an efficient constraint-based parser and a stochastic disambiguation model. The first evaluation metric 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 subset of the UPenn Wall Street Journal treebank. The other metric maps predicate-argument relations in LFG f-structures to dependency relations (henceforth DR annotations) as proposed by Carroll et al. (Carroll et al., 1999). For evaluation, these relations are matched against Carroll et al.'s gold standard which was manually annnotated on a subset of the Brown corpus. The parser plus stochastic disambiguator gives an F-measure of 79% (LFG) or 73% (DR) on the WSJ test set. This shows that the two evaluation schemes are similar in spirit, although accuracy is impaired systematically by mapping one annotation scheme to the other. A systematic loss of accuracy is incurred also by corpus variation: Training the stochastic disambiguation model on WSJ data and testing on Carroll et al.'s Brown corpus data yields an F-score of 74% (DR) for dependency-relation match. A variant of this measure comparable to the measure reported by Carroll et al. yields an F-measure of 76%. We examine divergences between annotation schemes aiming at a future improvement of methods for assessing parser quality.

1. Introduction

Recent years have seen increased interest in parsing systems that capture predicate-argument relations instead of mere phrase-structure representations. In aiming for this goal, considerable progress has been made by combining systems of hand-coded, linguistically fine-grained grammars with robustness techniques and stochastic disambiguation models. However, it can reasonably be argued that the standard evaluation procedure for stochastic parsing—precision and recall of matching labeled bracketing to section 23 of the UPenn Wall Street Journal (WSJ) treebank (Marcus et al., 1994)—is not appropriate for assessing the quality of parsers on matching predicate-argument relations. A new standard for evaluation on predicate-argument relations and for annotating a gold standard is needed.

In this paper we present a stochastic parsing model consisting of a broad-coverage Lexical-Functional Grammar (LFG), a constraint-based parser and a stochastic disambiguation model, and discuss the evaluation of this system on two distinct evaluation metrics for assessing the quality of the stochastic parsing model on matching predicate-argument relations. The first evaluation metric 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.

The other metric we employed maps predicate-argument relations in LFG f-structures to the dependency relations (henceforth the DR annotation scheme) proposed by Carroll et al. (Carroll et al., 1999). Evaluation with this metric measures the matches of these relations to Carroll et al.'s gold standard corpus.

Our parser plus stochastic disambiguator gives an F-measure of 79% (LFG) or 73% (DR) on the WSJ test set, showing that the two evaluation schemes are similar in spirit. However, accuracy is systematically impaired by mapping one annotation scheme to the other. A systematic loss of accuracy is incurred also by corpus variation: Training the stochastic disambiguation model on WSJ data and testing on Carroll et al.'s Brown corpus data gives a DR F-measure of 74% for matching dependency relations. For a direct comparison of our results with Carroll et al.'s system, we also computed an F-measure that does not distinguish different types of dependency relations. Under this measure we obtain 76% F-measure.

One goal of this paper is to highlight possible pitfalls and error sources in translating between different annotation schemes and gold standards. We believe that a thorough investigation of divergences in annotation schemes will facilitate a future standard for predicate-argument evaluation and annotation.

This paper is organized as follows. After introducing the grammar and parser used in this experiment, we describe in section 2. the robustness techniques employed to reach 100% grammar coverage on unseen WSJ text (in the sense of the proportion of sentences for which at least one analysis is found). Furthermore, we give in section 3. a short account of the stochastic model used for disambiguating LFG parses. Experiments on evaluating the combined system of parser and stochastic disambiguator on the two distinct evaluation measures and corpora are described in section 4.

2. Robust Parsing using LFG

2.1. A Broad-Coverage Lexical-Functional Grammar

The grammar used for this project has been 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. Each c-structure has at least one corresponding f-structure. 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 other lexicons and one for those not recognized. As such, most common and proper 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 such as prepositions.

For estimation and testing 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 and reliable for determining grammatical relations. The WSJ labels are given entries in a special LFG lexicon, and these entries 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 version 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, usually when it is a SUBJ or OBJ in one clause and also the SUBJ of an XCOMP complement. 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, presumably because it is the subject of a small clause complement. However, the LFG grammar treats it also 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 the LFG output from parsing its unlabeled, string-only counterpart is well-formed. Some other bracketing mismatches remain between this stripped down WSJ corpus and the LFG grammar; these are 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 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, just as in the standard grammar. Any substring 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 final capability of XLE that increases coverage of the standard plus fragment grammar on the WSJ corpus is a SKIMMING technique. Skimming is used to avoid timeouts and memory problems when parsing unusually difficult sentences in the corpus. 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, although it does not necessarily return the complete set of analyses. In parsing section 23, 7.2% of the sentences were skimmed; 26.1% of the skimmed sentences resulted in full parses, while 73.9% were fragment parses.

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

3. Discriminative Statistical Estimation from Partially Labeled Data

3.1. Exponential Probability Models on LFG Parses

The probability model we employed for stochastic disambiguation is the well-known family of exponential models. These models have already been applied successfully for disambiguation of various constraint-based grammars (LFG (Johnson et al., 1999), HPSG (Bouma et al., 2000), DCG (Osborne, 2000)).

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 employed 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, corresponding 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 directly 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 $X(y)$ 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 parameters 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 partial annotation, 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 adapted a conjugate gradient routine to our task (see Press (1992)), 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

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

[1] An earlier approach using partially labeled data for estimating stochastics parsers is Pereira and Schabes (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.

[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).

taken under consideration. From this set, sentences from 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 sentence 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 of 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 150 sentences which received only full parses, out of which the most plausible one was selected by manual inspection.

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]. Both the LFG and DR annotation schemes are discussed in more detail below, as is a mapping from LFG f-structures to DR annotations.

Gold standard annotation of 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.

Performance on the LFG-annotated WSJ test set was measured using both the LFG and DR metrics, thanks to the LFG-to-DR annotation mapping. Performance on the DR-annotated Brown test set was only measured using the DR metric, owing to the absence of an inverse map from DR to LFG annotations.

Results: In our evaluation we report F-measures for the respective types of annotation, LFG or DR, and for three types of parse selection, (i) *lower bound*: random choice of a parse from the set of analyses, (ii) *upper bound*: selection of the parse with the best F-measure 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-measures is defined as $2 \times precision \times recall/(precision + recall)$.

[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`.

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

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

	LFG	DR
upper bound	84.7	80.7
stochastic	78.7	72.9
lower bound	75.0	68.8
error reduction	38	35

disambiguation performance can be illustrated by breaking down the F-measures according to whether the parser yields full parses or FRAGMENT or SKIMMED parses or both 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-measures broken down according to classes of parse quality are recorded in the following rows. The first column shows F-measures for all parses in the test set, as in Table 1, the second column shows best F-measures when restricting attention to examples which receive only full parses. The third column reports F-measurs 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. Since most results on predicate-argument matching have been reported for length-restricted test sets (20–30 words), we also provide for comparison results for a subset of 500 sentences in our sample which had less than 25 words. These results are reported in Table 3.

Table 3: Disambiguation results on 500 examples restricted to < 25 words randomly selected from section 23 of the WSJ treebank using LFG and DR measures.

	LFG	DR
upper bound	88.0	85.4
stochastic	82.8	77.5
lower bound	78.0	72.6
error reduction	42	38

Results of the evaluation on Carroll et al.'s Brown test set are given in Tables 4 and 5. Table 4 presents an analysis of evaluation results according to parse-quality for the DR measure applied to the Brown corpus test set. In Table 5 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 depedency 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 be-

Table 2: LFG F-measures broken down according to parse quality for the 700 WSJ test examples.

	all	full	non-full	fragments	skimmed	skimmed fragments
% of test set	100	74.7	25.3	20.4	1.4	3.4
upper bound	84.7	91.3	69.8	72.0	73.1	60.5
stochastic	78.8	84.6	65.2	67.4	67.8	55.9
lower bound	75.0	80.1	63.9	65.9	66.2	55.3

Table 4: DR F-measures broken down according to parse quality for the 500 Brown test examples.

	all	full	non-full	fragments	skimmed	skimmed fragments
% of test set	100	79.6	20.4	20.0	2.0	1.6
upper bound	79.6	84.0	65.2	65.2	55.5	52.9
stochastic	73.7	77.6	61.1	61.0	52.3	49.4
lower bound	70.8	74.4	58.8	58.7	50.8	48.3

tween arguments and modifiers in a small number of cases.

Table 5: Disambiguation results on 500 Brown corpus examples using DO measure and DR measures.

	DO	DR
upper bound	81.6	79.6
stochastic	75.8	73.7
lower bound	72.9	70.8
error reduction	33	34

5. Comparison of Evaluation Metrics

Tables 1 and 3 point to systematically lower F-scores under the DR measure than under the LFG measure, though both indicate similar reductions in error rate due to stochastic disambiguation.

5.1. LFG Evaluation Metric

The LFG evaluation metric is based on the comparison of 'preds-only' f-structures. A preds-only f-structure is a subset of a full f-structure that strips out grammatical attributes (e.g. tense, case, number) that are not directly relevant to predicate-argument structure. More precisely, a preds-only f-structure removes all paths through the f-structure that do not end in a PRED attribute. Figures 1 and 2 illustrate the difference between the full and preds-only f-structures for one parse of the sentence *Meridian will pay a premium of $30.5 million to assume a deposit of $2 billion.* As this example shows, the preds-only f-structure lacks some semantically important information present in the full f-structure, e.g. the marking of future tense, the marking of a purpose clause, and the attribute showing that *a deposit* is an indefinite.

Figure 2 also shows the set of individual feature specifications that define the preds-only f-structure. The first property indicates that the f-structure denoted by `n0` has the semantic form `sf(pay,i15,[n5,n3],[])` as the value of its PRED attribute. `pay` is the predicate, `i15` is a lexical id, `[n5,n3]` a list of f-structure nodes serving as thematic arguments, and `[]` an (empty) list of non-thematic arguments. The grammatical roles associated with thematic and non-thematic arguments are identified by the corresponding `subj`, `obj`, etc., predicates. In this experiment, we measured precision and recall by matching at the granularity of these individual features.

The matching algorithm attempts to find the maximum number of features that can be matched between two structures. It proceeds in a stratified manner, first maximizing the matches between attributes like `pred`, `adjunct` and `in_set`, and then maximizing the matches of any remaining attributes.

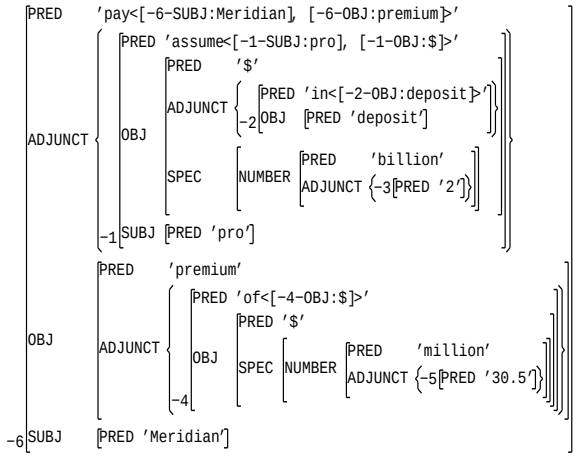
5.2. Comparison with DR Metric

As a brief review (see Carroll et al. (1999) for more detail), the DR annotation for our example sentence (obtained via the mapping described below) is

(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)

Some obvious points of comparison with the f-structure features are: (i) The DR annotation encodes some information, e.g. the 'detmod' relation, that is not encoded in preds-only f-structures (though it is encoded in full f-structures). (ii) Different occurrences of the same word (e.g. "$") are distinguished via different lexical ids in the LFG representation but not in the DR annotations so that correctly matching DR relations can be problematic. (iii) The DR annotation has 12 relations instead of the 34 feature-specifications. This is because a given predicate-argument relation in the f-structure is broken down into several different feature-specifications. For example, the DR 'mod' relation involves an f-structure path through an ADJUNCT, IN_SET and two PRED attributes; the DR 'subj' relation is a combination of an f-structure PRED and SUBJ attribute. Thus the LFG metric is more sensitive to fine-grained aspects of predicate-

Figure 1: Full f-structure



pred(n0,sf(pay,i15,[n5,n3],[])) pred(n5,sf(Meridian,il4,[],[]))
pred(n3,sf(premium,i18,[],[])) pred(n19,sf('2',i70,[],[]))
pred(n28,sf('30.5',i26,[],[])) pred(n7,sf(assume,i64,[n8,n9],[]))
pred(n8,sf(pro,i107,[],[])) pred(n9,sf($,i67,[],[]))
pred(n17,sf(billion,i71,[],[])) pred(n11,sf(in,i84,[n12],[]))
pred(n12,sf(deposit,i86,[],[])) pred(n4,sf(million,i27,[],[]))
pred(n23,sf(of,i21,[n24],[])) pred(n24,sf($,i23,[],[]))
adjunct(n0,n2) in_set(n7,n2)
adjunct(n9,n14) in_set(n11,n14)
adjunct(n17,n18) in_set(n19,n18)
adjunct(n3,n20) in_set(n23,n20)
adjunct(n4,n31) in_set(n28,n31)
subj(n0,n5) subj(n7,n8)
obj(n0,n3) obj(n7,n9) obj(n11,n12) obj(n23,n24)
number(n16,n17) number(n26,n4) spec(n9,n16) spec(n24,n26)

Figure 2: Preds-only f-structure: graphical & clausal representation as produced by XLE

argument relations. However, it imposes a greater penalty than DR on a modifier that is misattached to something that does not have any other modifiers. The LFG measure counts both an extra ADJUNCT feature and an extra IN_SET feature as mismatches, whereas DR only counts a single mismatched MOD. Conversely, LFG gives more credit for getting the singleton attachments correct. Similarly for argument structure. The LFG metric penalizes getting arguments wrong, counting both a PRED and a grammatical relation mismatch, but conversely gives more credit if the argument structure is exactly right.

5.3. Mapping F-structures to DR Annotations

The DR evaluation metric matches the dependency relations provided by the Carroll et al. gold standard with relations determined from information contained in the LFG representations. This enables us to measure the accuracy of our system with a separately defined predicate-argument-oriented standard and to compare our results to other systems that may use the same metric (at this point, perhaps only the Carroll et al. grammar/parser). The DR metric also enables a cross-validation assessment of the LFG-derived predicate-argument measure.

Carroll and Briscoe provide conveniently downloadable files containing the raw input sentences and the corresponding sets of gold standard dependency relations. We assumed it would be relatively straightforward to run the sentences through our system and extract dependency relations that could be compared to the gold standard. But for reasons that ranged from the ridiculous to the sublime, this turned out to be a surprisingly difficult task. One of the lessons learned from this experiment is that even at the level of abstract dependencies it is still very hard to create a standard that does not incorporate unintended framework-specific idiosyncracies.

One set of problems arose from the way the sentences are recorded in the input file. The 'raw' sentences are not formed as they would appear in natural text. They are pro-

vided instead as pre-tokenized strings, with punctuation split off by spaces from surrounding words. Thus commas and periods stand as separate tokens and *I'm* and *clients' guilt* show up as *I 'm* and *clients ' guilt*. This preprocessed format may be helpful for parsing systems that embody this particular set of tokenizing conventions or that learn (a la tree bank grammars) from the data at hand. But our system includes a hand-written finite-state tokenizer that is tightly integrated with our grammar and lexicon, and it is designed to operate on text that conforms to normal typographical conventions. It provides less accurate guesses when text is ill-formed in this way, for example, introducing an ambiguity as to whether the quote in *clients ' guilt* is attached as a genitive marker to the left or as an open quote to the right. Another peculiar and troublesome feature of the raw text is that some non-linguistic elements such as chemical formulas are replaced by the meta-symbol <*formul*>; our tokenizer splits this up at the angle brackets and tries to guess a meaning for the word *formul* surrounded by brackets. Faced with these low-level peculiarities, our first step in the evaluation was to edit the raw text as best we could back into normal English.

The gold standard file presented another set of relatively low-level incompatibilities that resulted in spurious mismatches that were somewhat harder to deal with. First, the input sentences conform to American spelling conventions but the head-words in the gold standard relations use British spelling (*neighbor* is coded as *neighbour*). Second, in the gold standard the head-words are converted to their citation forms (e.g. "*walking*" in the text appears as *walk* in the relations). Generally these match the head-words that are easily read from the LFG f-structures, but there are many discrepancies that had to be tracked down. For example, our f-structures do not convert *should* to *shall*, as the gold standard does, whereas we do convert *himself* to *he* (with a reflexive feature) while the gold standard leaves it as *himself*. We ended up creating by trial-and-error a coercion table for this test set so that we could properly match different manifestations of the same head.

The experiment revealed some higher-level conceptual issues. In LFG it is the f-structure rather than the c-structure that most closely encodes the properties on which a non-tree, dependency-oriented evaluation should be based. So we defined our task to be the construction of a routine for reading dependencies from the f-structure alone. It turns out, however, that the Carroll et al. dependencies encode a mixture of superficial phrase-structure properties in addition to underlying dependencies, and it proved a challenge to recreate all the information relevant to a match from the f-structure alone. For example, our f-structures do not represent the categories (NP, S) of the phrases that correspond to the functions, but the gold standard dependencies make tree-based distinctions between non-clausal (e.g. NP) subjects, clausal (e.g. sentential) subjects, and open-complement (VP) subjects. We avoided this kind of discrepancy by neutralizing these distinctions in the gold standard prior to making any comparisons. As another example, our English grammar decodes English auxiliary sequences into features such as PERFECT, PROGRESSIVE, and PASSIVE while the gold standard provides a set of AUX relations that represent the left-to-right order in which *have* and *be* appeared in the original sentence. To obtain the intuitively correct matches, our mapping routine in effect had to simulate a small part of an English generator that decodes our features into their typical left-to-right ordering. In at least one case we simply gave up—it was too hard to figure out under which conditions there might have been do-support in the original string; instead, we removed the few aux-do relations from the gold standard before comparing.

There were a number of situations where it was difficult to determine exactly the gold standard coding conventions either from the documentation or from the examples in the gold standard file. Some of the confusions were resolved by personal communication with Carroll and Briscoe, leading in some cases to the correction of errors in the standard or to the clarification of principles. We discovered for some phenomena that there were simple differences of opinion of how a relation should be annotated. The corpus contains many parentheticals, for example, whose proper attachment is generally determined by extrasyntactic, discourse-level considerations. The default in the LFG grammar is to associate parentheticals at the clause-level whereas the Carroll-Briscoe gold standard tends to associate them with the constituent immediately to the left—a constituent that we cannot identify from the f-structure alone. As other examples, there are still some mysteries about whether and how unexpressed subjects of open-complements are to be encoded and whether and how the head of a relative clause appears in a within-clause dependency.

With considerable effort we solved most but not all of these cross-representation mapping problems, as attested by the relatively high F-scores we have reported. Our current results probably understate to a certain extent our true degree of matching, but the relative differences between sentences using the DR measure are quite informative. A low F-score is an accurate indication that we did not obtain the correct parse. For F-scores above 90 but below 100 it is often the case that we found exactly the right parse but our mapping routine could not produce all the proper relations.

6. Discussion

The general conclusion to draw from our results is that the two metrics, LFG and DR, show broadly similar behavior, for the upper bounds, for the lower bounds, and for the reduction in error relative to the upper bound brought about by the stochastic model. The correlation between the upper bound F-scores for the LFG and DR measures on the WSJ test set is .89. 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.[4] Breaking down evaluation results according to parse quality shows that irrespective of evaluation measure and corpus around 5% overall per-

[4] 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.

formance is lost due to non-full parses, i.e. FRAGMENT or SKIMMED parses or both.

While disambiguation performance of around 79% F-score on WSJ data seems promising, from one perspective it only offers a 4% 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 (38% 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.

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 at the moment. 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-measure of 75.1% for a DO evaluation that ignores predicate labels but counts dependencies only. Under this measure, our system of parser and stochastic disambiguator achieves 75.8% F-measure. A further point of comparison is the parsing system presented by Bouma et al. (2000). They report comparable relations on lower bounds and upper bounds for their constraint-based parsing systems. On test corpora of a few hundred sentences of up to 20 words an upper bound of 83.7% F-score and a lower bound of 59% is reported; the best disambiguation models achieves 75% F-score.

7. 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.

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.

Miles Osborne. 2000. Estimation of stochastic attribute-value grammars using an informative sample. In *Proceedings of the 18th International Conference on Computational Linguistics (COLING 2000)*, Saarbrücken.

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.

William H. Press, Saul A. Teukolsky, Willam T. Vetterling, and Brian P. Flannery. 1992. *Numerical Recipes in C: The Art of Scientific Computing*. Cambridge University Press, New York.

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.