Evaluating the Accuracy of an Unlexicalized Statistical Parser on the PARC DepBank

Ted Briscoe
Computer Laboratory
University of Cambridge

John Carroll
School of Informatics
University of Sussex

Abstract

We evaluate the accuracy of an unlexicalized statistical parser, trained on 4K treebanked sentences from balanced data and tested on the PARC DepBank. We demonstrate that a parser which is competitive in accuracy (without sacrificing processing speed) can be quickly tuned without reliance on large in-domain manually-constructed treebanks. This makes it more practical to use statistical parsers in applications that need access to aspects of predicate-argument structure. The comparison of systems using DepBank is not straightforward, so we extend and validate DepBank and highlight a number of representation and scoring issues for relational evaluation schemes.

1 Introduction

Considerable progress has been made in accurate statistical parsing of realistic texts, yielding rooted, hierarchical and/or relational representations of full sentences. However, much of this progress has been made with systems based on large lexicalized probabilistic context-free like (PCFG-like) models trained on the Wall Street Journal (WSJ) subset of the Penn TreeBank (PTB). Evaluation of these systems has been mostly in terms of the PARSEVAL scheme using tree similarity measures of (labelled) precision and recall and crossing bracket rate applied to section 23 of the WSJ PTB. (See e.g. Collins (1999) for detailed exposition of one such very fruitful line of research.)

We evaluate the comparative accuracy of an unlexicalized statistical parser trained on a smaller treebank and tested on a subset of section 23 of the WSJ using a relational evaluation scheme. We demonstrate that a parser which is competitive in accuracy (without sacrificing processing speed) can be quickly developed without reliance on large in-domain manually-constructed treebanks. This makes it more practical to use statistical parsers in diverse applications needing access to aspects of predicate-argument structure.

We define a lexicalized statistical parser as one which utilizes probabilistic parameters concerning lexical subcategorization and/or bilexical relations over tree configurations. Current lexicalized statistical parsers developed, trained and tested on PTB achieve a labelled F_1-score – the harmonic mean of labelled precision and recall – of around 90%. Klein and Manning (2003) argue that such results represent about 4% absolute improvement over a carefully constructed unlexicalized PCFG-like model trained and tested in the same manner.[1] Gildea (2001) shows that WSJ-derived bilexical parameters in Collins' (1999) Model 1 parser contribute less than 1% to parse selection accuracy when test data is in the same domain, and yield no improvement for test data selected from the Brown Corpus. Bikel (2004) shows that, in Collins' (1999) Model 2, bilexical parameters contribute less than 0.5% to accuracy on in-domain data while lexical subcategorization-like parameters contribute just over 1%.

Several alternative relational evaluation schemes have been developed (e.g. Carroll *et al.*, 1998; Lin, 1998). However, until recently, no WSJ data has been carefully annotated to support relational evaluation. King *et al.* (2003) describe the PARC 700 Dependency Bank (hereinafter *DepBank*), which consists of 700 WSJ sentences randomly drawn from section 23. These sentences have been annotated with syntactic features and with bilexical head-dependent relations derived from the F-structure representation of Lexical Functional Grammar (LFG). DepBank facilitates

[1] Klein and Manning retained some functional tag information from PTB, so it could be argued that their model remains 'mildly' lexicalized since functional tags encode some subcategorization information.

Proceedings of the COLING/ACL 2006 Main Conference Poster Sessions, pages 41–48,
Sydney, July 2006.

comparison of PCFG-like statistical parsers developed from the PTB with other parsers whose output is not designed to yield PTB-style trees, using an evaluation which is closer to the protypical parsing task of recovering predicate-argument structure.

Kaplan *et al.* (2004) compare the accuracy and speed of the PARC XLE Parser to Collins' Model 3 parser. They develop transformation rules for both, designed to map native output to a subset of the features and relations in DepBank. They compare performance of a grammatically cut-down and complete version of the XLE parser to the publically available version of Collins' parser. One fifth of DepBank is held out to optimize the speed and accuracy of the three systems. They conclude from the results of these experiments that the cut-down XLE parser is two-thirds the speed of Collins' Model 3 but 12% more accurate, while the complete XLE system is 20% more accurate but five times slower. F_1-score percentages range from the mid- to high-70s, suggesting that the relational evaluation is harder than PARSEVAL.

Both Collins' Model 3 and the XLE Parser use lexicalized models for parse selection trained on the rest of the WSJ PTB. Therefore, although Kaplan *et al.* demonstrate an improvement in accuracy at some cost to speed, there remain questions concerning viability for applications, at some remove from the financial news domain, for which substantial treebanks are not available. The parser we deploy, like the XLE one, is based on a manually-defined feature-based unification grammar. However, the approach is somewhat different, making maximal use of more generic structural rather than lexical information, both within the grammar and the probabilistic parse selection model. Here we compare the accuracy of our parser with Kaplan *et al.*'s results, by repeating their experiment with our parser. This comparison is not straightforward, given both the system-specific nature of some of the annotation in DepBank and the scoring reported. We, therefore, extend DepBank with a set of grammatical relations derived from our own system output and highlight how issues of representation and scoring can affect results and their interpretation.

In §2, we describe our development methodology and the resulting system in greater detail. §3 describes the extended Depbank that we have developed and motivates our additions. §2.4 discusses how we trained and tuned our current system and describes our limited use of information derived from WSJ text. §4 details the various experiments undertaken with the extended DepBank and gives detailed results. §5 discusses these results and proposes further lines of research.

2 Unlexicalized Statistical Parsing

2.1 System Architecture

Both the XLE system and Collins' Model 3 preprocess textual input before parsing. Similarly, our baseline system consists of a pipeline of modules. First, text is tokenized using a deterministic finite-state transducer. Second, tokens are part-of-speech and punctuation (PoS) tagged using a 1st-order Hidden Markov Model (HMM) utilizing a lexicon of just over 50K words and an unknown word handling module. Third, deterministic morphological analysis is performed on each token-tag pair with a finite-state transducer. Fourth, the lattice of lemma-affix-tags is parsed using a grammar over such tags. Finally, the *n*-best parses are computed from the parse forest using a probabilistic parse selection model conditioned on the structural parse context. The output of the parser can be displayed as syntactic trees, and/or factored into a sequence of bilexical grammatical relations (GRs) between lexical heads and their dependents.

The full system can be extended in a variety of ways – for example, by pruning PoS tags but allowing multiple tag possibilities per word as input to the parser, by incorporating lexical subcategorization into parse selection, by computing GR weights based on the proportion and probability of the *n*-best analyses yielding them, and so forth – broadly trading accuracy and greater domain-dependence against speed and reduced sensitivity to domain-specific lexical behaviour (Briscoe and Carroll, 2002; Carroll and Briscoe, 2002; Watson *et al.*, 2005; Watson, 2006). However, in this paper we focus exclusively on the baseline unlexicalized system.

2.2 Grammar Development

The grammar is expressed in a feature-based, unification formalism. There are currently 676 phrase structure rule schemata, 15 feature propagation rules, 30 default feature value rules, 22 category expansion rules and 41 feature types which together define 1124 compiled phrase structure rules in which categories are represented as sets of fea-

tures, that is, attribute-value pairs, possibly with variable values, possibly bound between mother and one or more daughter categories. 142 of the phrase structure schemata are manually identified as peripheral rather than core rules of English grammar. Categories are matched using fixed-arity term unification at parse time.

The lexical categories of the grammar consist of feature-based descriptions of the 149 PoS tags and 13 punctuation tags (a subset of the CLAWS tagset, see e.g. Sampson, 1995) which constitute the preterminals of the grammar. The number of distinct lexical categories associated with each preterminal varies from 1 for some function words through to around 35 as, for instance, tags for main verbs are associated with a VSUBCAT attribute taking 33 possible values. The grammar is designed to enumerate possible valencies for predicates by including separate rules for each pattern of possible complementation in English. The distinction between arguments and adjuncts is expressed by adjunction of adjuncts to maximal projections ($XP \rightarrow XP\ Adjunct$) as opposed to government of arguments (i.e. arguments are sisters within *X1* projections; $X1 \rightarrow X0\ Arg1 \ldots ArgN$).

Each phrase structure schema is associated with one or more GR specifications which can be conditioned on feature values instantiated at parse time and which yield a rule-to-rule mapping from local trees to GRs. The set of GRs associated with a given derivation define a connected, directed graph with individual nodes representing lemma-affix-tags and arcs representing named grammatical relations. The encoding of this mapping within the grammar is similar to that of F-structure mapping in LFG. However, the connected graph is not constructed and completeness and coherence constraints are not used to filter the phrase structure derivation space.

The grammar finds at least one parse rooted in the start category for 85% of the Susanne treebank, a 140K word balanced subset of the Brown Corpus, which we have used for development (Sampson, 1995). Much of the remaining data consists of phrasal fragments marked as independent text sentences, for example in dialogue. Grammatical coverage includes the majority of construction types of English, however the handling of some unbounded dependency constructions, particularly comparatives and equatives, is limited because of the lack of fine-grained subcategorization information in the PoS tags and by the need to balance depth of analysis against the size of the derivation space. On the Susanne corpus, the geometric mean of the number of analyses for a sentence of length n is 1.31^n. The microaveraged F_1-score for GR extraction on held-out data from Susanne is 76.5% (see section 4.2 for details of the evaluation scheme).

The system has been used to analyse about 150 million words of English text drawn primarily from the PTB, TREC, BNC, and Reuters RCV1 datasets in connection with a variety of projects. The grammar and PoS tagger lexicon have been incrementally improved by manually examining cases of parse failure on these datasets. However, the effort invested amounts to a few days' effort for each new dataset as opposed to the main grammar development effort, centred on Susanne, which has extended over some years and now amounts to about 2 years' effort (see Briscoe, 2006 for further details).

2.3 Parser

To build the parsing module, the unification grammar is automatically converted into an atomic-categoried context free 'backbone', and a non-deterministic LALR(1) table is constructed from this, which is used to drive the parser. The residue of features not incorporated into the backbone are unified on each rule application (reduce action). In practice, the parser takes average time roughly quadratic in the length of the input to create a packed parse forest represented as a graph-structured stack. The statistical disambiguation phase is trained on Susanne treebank bracketings, producing a probabilistic generalized LALR(1) parser (e.g. Inui *et al.*, 1997) which associates probabilities with alternative actions in the LR table.

The parser is passed as input the sequence of most probable lemma-affix-tags found by the tagger. During parsing, probabilities are assigned to subanalyses based on the the LR table actions that derived them. The n-best (i.e. most probable) parses are extracted by a dynamic programming procedure over subanalyses (represented by nodes in the parse forest). The search is efficient since probabilities are associated with single nodes in the parse forest and no weight function over ancestor or sibling nodes is needed. Probabilities capture structural context, since nodes in

the parse forest partially encode a configuration of the graph-structured stack and lookahead symbol, so that, unlike a standard PCFG, the model discriminates between derivations which only differ in the order of application of the same rules and also conditions rule application on the PoS tag of the lookahead token.

When there is no parse rooted in the start category, the parser returns a connected sequence of partial parses which covers the input based on subanalysis probability and a preference for longer and non-lexical subanalysis combinations (e.g. Kiefer *et al.*, 1999). In these cases, the GR graph will not be fully connected.

2.4 Tuning and Training Method

The HMM tagger has been trained on 3M words of balanced text drawn from the LOB, BNC and Susanne corpora, which are available with hand-corrected CLAWS tags. The parser has been trained from 1.9K trees for sentences from Susanne that were interactively parsed to manually obtain the correct derivation, and also from 2.1K further sentences with unlabelled bracketings derived from the Susanne treebank. These bracketings guide the parser to one or possibly several closely-matching derivations and these are used to derive probabilities for the LR table using (weighted) Laplace estimation. Actions in the table involving rules marked as peripheral are assigned a uniform low prior probability to ensure that derivations involving such rules are consistently lower ranked than those involving only core rules.

To improve performance on WSJ text, we examined some parse failures from sections other than section 23 to identify patterns of consistent failure. We then manually modified and extended the grammar with a further 6 rules, mostly to handle cases of indirect and direct quotation that are very common in this dataset. This involved 3 days' work. Once completed, the parser was retrained on the original data. A subsequent limited inspection of top-ranked parses led us to disable 6 existing rules which applied too freely to the WSJ text; these were designed to analyse auxiliary ellipsis which appears to be rare in this genre. We also catalogued incorrect PoS tags from WSJ parse failures and manually modified the tagger lexicon where appropriate. These modifications mostly consisted of adjusting lexical probabilities of extant entries with highly-skewed distributions. We also added some tags to extant entries for infrequent words. These modifications took a further day. The tag transition probabilities were not reestimated. Thus, we have made no use of the PTB itself and only limited use of WSJ text.

This method of grammar and lexicon development incrementally improves the overall performance of the system averaged across all the datasets that it has been applied to. It is very likely that retraining the PoS tagger on the WSJ and retraining the parser using PTB would yield a system which would perform more effectively on DepBank. However, one of our goals is to demonstrate that an unlexicalized parser trained on a modest amount of annotated text from other sources, coupled to a tagger also trained on generic, balanced data, can perform competitively with systems which have been (almost) entirely developed and trained using PTB, whether or not these systems deploy hand-crafted grammars or ones derived automatically from treebanks.

3 Extending and Validating DepBank

DepBank was constructed by parsing the selected section 23 WSJ sentences with the XLE system and outputting syntactic features and bilexical relations from the F-structure found by the parser. These features and relations were subsequently checked, corrected and extended interactively with the aid of software tools (King *et al.*, 2003).

The choice of relations and features is based quite closely on LFG and, in fact, overlaps substantially with the GR output of our parser. Figure 1 illustrates some DepBank annotations used in the experiment reported by Kaplan *et al.* and our hand-corrected GR output for the example *Ten of the nation's governors meanwhile called on the justices to reject efforts to limit abortions*. We have kept the GR representation simpler and more readable by suppressing lemmatization, token numbering and PoS tags, but have left the DepBank annotations unmodified.

The example illustrates some differences between the schemes. For instance, the **subj** and **ncsubj** relations overlap as both annotations contain such a relation between *call(ed)* and *Ten*), but the GR annotation also includes this relation between *limit* and *effort(s)* and *reject* and *justice(s)*, while DepBank links these two verbs to a variable **pro**. This reflects a difference of philosophy about

DepBank:
```
obl(call~0, on~2)
stmt_type(call~0, declarative)
subj(call~0, ten~1)
tense(call~0, past)
number_type(ten~1, cardinal)
obl(ten~1, governor~35)
obj(on~2, justice~30)
obj(limit~7, abortion~15)
subj(limit~7, pro~21)
obj(reject~8, effort~10)
subj(reject~8, pro~27)
adegree(meanwhile~9, positive)
num(effort~10, pl)
xcomp(effort~10, limit~7)
```

GR:
```
(ncsubj called Ten _)
(ncsubj reject justices _)
(ncsubj limit efforts _)
(iobj called on)
(xcomp to called reject)
(dobj reject efforts)
(xmod to efforts limit)
(dobj limit abortions)
(dobj on justices)
(det justices the)
(ta bal governors meanwhile)
(ncmod poss governors nation)
(iobj Ten of)
(dobj of governors)
(det nation the)
```

Figure 1: DepBank and GR annotations.

resolution of such 'understood' relations in different constructions. Viewed as output appropriate to specific applications, either approach is justifiable. However, for evaluation, these DepBank relations add little or no information not already specified by the **xcomp** relations in which these verbs also appear as dependents. On the other hand, DepBank includes an **adjunct** relation between *meanwhile* and *call(ed)*, while the GR annotation treats *meanwhile* as a text adjunct (**ta**) of *governors*, delimited by balanced commas, following Nunberg's (1990) text grammar but conveying less information here.

There are also issues of incompatible tokenization and lemmatization between the systems and of differing syntactic annotation of similar information, which lead to problems mapping between our GR output and the current DepBank. Finally, differences in the linguistic intuitions of the annotators and errors of commission or omission on both sides can only be uncovered by manual comparison of output (e.g. **xmod** vs. **xcomp** for *limit efforts* above). Thus we reannotated the DepBank sentences with GRs using our current system, and then corrected and extended this annotation utilizing a software tool to highlight differences between the extant annotations and our own.[2] This exercise, though time-consuming, uncovered problems in both annotations, and yields a doubly-annotated and potentially more valuable resource in which annotation disagreements over complex attachment decisions, for instance, can be inspected.

The GR scheme includes one feature in DepBank (**passive**), several splits of relations in DepBank, such as **adjunct**, adds some of DepBank's featural information, such as **subord_form**, as a subtype slot of a relation (**ccomp**), merges DepBank's **obl**ique with **iobj**, and so forth. But it does not explicitly include all the features of DepBank or even of the reduced set of semantically-relevant features used in the experiments and evaluation reported in Kaplan *et al.*. Most of these features can be computed from the full GR representation of bilexical relations between numbered lemma-affix-tags output by the parser. For instance, **num** features, such as the plurality of *justices* in the example, can be computed from the full **det** GR `(det justice+s_NN2:4 the_AT:3)` based on the CLAWS tag (NN**2** indicating 'plural') selected for output. The few features that cannot be computed from GRs and CLAWS tags directly, such as **stmt_type**, could be computed from the derivation tree.

4 Experiments

4.1 Experimental Design

We selected the same 560 sentences as test data as Kaplan *et al.*, and all modifications that we made to our system (see §2.4) were made on the basis of (very limited) information from other sections of WSJ text.[3] We have made no use of the further 140 held out sentences in DepBank. The results we report below are derived by choosing the most probable tag for each word returned by the PoS tagger and by choosing the unweighted GR set returned for the most probable parse with no lexical information guiding parse ranking.

4.2 Results

Our parser produced rooted sentential analyses for 84% of the test items; actual coverage is higher

[2] The new version of DepBank along with evaluation software is included in the current RASP distribution: www.informatics.susx.ac.uk/research/nlp/rasp

[3] The PARC group kindly supplied us with the experimental data files they used to facilitate accurate reproduction of this experiment.

Relation	Precision	Recall	F_1	*P R F_1 Relation*
mod	75.4	71.2	73.3	
ncmod	72.9	67.9	70.3	
xmod	47.7	45.5	46.6	
cmod	51.4	31.6	39.1	
pmod	30.8	33.3	32.0	
det	88.7	91.1	89.9	
arg_mod	71.9	67.9	69.9	
arg	76.0	73.4	74.6	
subj	80.1	66.6	72.7	*73 73 73*
ncsubj	80.5	66.8	73.0	
xsubj	50.0	28.6	36.4	
csubj	20.0	50.0	28.6	
subj_or_dobj	82.1	74.9	78.4	
comp	74.5	76.4	75.5	
obj	78.4	77.9	78.1	
dobj	83.4	81.4	82.4	*75 75 75 obj*
obj2	24.2	38.1	29.6	*42 36 39 obj-theta*
iobj	68.2	68.1	68.2	*64 83 72 obl*
clausal	63.5	71.6	67.3	
xcomp	75.0	76.4	75.7	*74 73 74*
ccomp	51.2	65.6	57.5	*78 64 70 comp*
pcomp	69.6	66.7	68.1	
aux	92.8	90.5	91.6	
conj	71.7	71.0	71.4	*68 62 65*
ta	39.1	48.2	43.2	
passive	93.6	70.6	80.5	*80 83 82*
adegree	89.2	72.4	79.9	*81 72 76*
coord_form	92.3	85.7	88.9	*92 93 93*
num	92.2	89.8	91.0	*86 87 86*
number_type	86.3	92.7	89.4	*96 95 96*
precoord_form	100.0	16.7	28.6	*100 50 67*
pron_form	92.1	91.9	92.0	*88 89 89*
prt_form	71.1	58.7	64.3	*72 65 68*
subord_form	60.7	48.1	53.6	
macroaverage	69.0	63.4	66.1	
microaverage	81.5	78.1	79.7	*80 79 79*

Table 1: Accuracy of our parser, and where roughly comparable, the XLE as reported by King *et al.*

than this since some of the test sentences are elliptical or fragmentary, but in many cases are recognized as single complete constituents. Kaplan *et al.* report that the complete XLE system finds rooted analyses for 79% of section 23 of the WSJ but do not report coverage just for the test sentences. The XLE parser uses several performance optimizations which mean that processing of subanalyses in longer sentences can be curtailed or preempted, so that it is not clear what proportion of the remaining data is outside grammatical coverage.

Table 1 shows accuracy results for each individual relation and feature, starting with the GR bilexical relations in the extended DepBank and followed by most DepBank features reported by Kaplan *et al.*, and finally overall macro- and microaverages. The macroaverage is calculated by taking the average of each measure for each individual relation and feature; the microaverage measures are calculated from the counts for all relations and features.[4] Indentation of GRs shows degree of specificity of the relation. Thus, **mod** scores are microaveraged over the counts for the five fully specified modifier relations listed immediately after it in Table 1. This allows comparison of overall accuracy on modifiers with, for instance overall accuracy on **arg**uments. Figures in italics to the right are discussed in the next section.

Kaplan *et al.*'s microaveraged scores for Collins' Model 3 and the cut-down and complete versions of the XLE parser are given in Table 2, along with the microaveraged scores for our parser from Table 1. Our system's accuracy results (evaluated on the reannotated DepBank) are better than those for Collins and the cut-down XLE, and very similar overall to the complete XLE (evaluated on DepBank). Speed of processing is also very competitive.[5] These results demonstrate that a statistical parser with roughly state-of-the-art accuracy can be constructed without the need for large in-domain treebanks. However, the performance of the system, as measured by microraveraged F_1-score on GR extraction alone, has declined by 2.7% over the held-out Susanne data, so even the unlexicalized parser is by no means domain-independent.

4.3 Evaluation Issues

The DepBank **num** feature on nouns is evaluated by Kaplan *et al.* on the grounds that it is semantically-relevant for applications. There are over 5K **num** features in DepBank so the overall microaveraged scores for a system will be significantly affected by accuracy on **num**. We expected our system, which incorporates a tagger with good empirical (97.1%) accuracy on the test data, to recover this feature with 95% accuracy or better, as it will correlate with tags NNx1 and NNx2 (where 'x' represents zero or more capitals in the CLAWS

[4] We did not compute the remaining DepBank features **stmt_type**, **tense**, **prog** or **perf** as these rely on information that can only be extracted from the derivation tree rather than the GR set.

[5] Processing time for our system was 61 seconds on one 2.2GHz Opteron CPU (comprising tokenization, tagging, morphology, and parsing, including module startup overheads). Allowing for slightly different CPUs, this is 2.5–10 times faster than the Collins and XLE parsers, as reported by Kaplan *et al.*

System	Eval corpus	Precision	Recall	F_1
Collins	DepBank	78.3	71.2	74.6
Cut-down XLE	DepBank	79.1	76.2	77.6
Complete XLE	DepBank	79.4	79.8	79.6
Our system	DepBank/GR	81.5	78.1	79.7

Table 2: Microaveraged overall scores from Kaplan *et al.* and for our system.

tagset). However, DepBank treats the majority of prenominal modifiers as adjectives rather than nouns and, therefore, associates them with an **adegree** rather than a **num** feature. The PoS tag selected depends primarily on the relative lexical probabilities of each tag for a given lexical item recorded in the tagger lexicon. But, regardless of this lexical decision, the correct GR is recovered, and neither **adegree(positive)** or **num(sg)** add anything semantically-relevant when the lexical item is a nominal premodifier. A strategy which only provided a **num** feature for nominal heads would be both more semantically-relevant and would also yield higher precision (95.2%). However, recall (48.4%) then suffers against DepBank as noun premodifiers have a **num** feature. Therefore, in the results presented in Table 1 we have not counted cases where either DepBank or our system assign a premodifier **adegree(positive)** or **num(sg)**.

There are similar issues with other DepBank features and relations. For instance, the form of a subordinator with clausal complements is annotated as a relation between verb and subordinator, while there is a separate **comp** relation between verb and complement head. The GR representation adds the subordinator as a subtype of **ccomp** recording essentially identical information in a single relation. So evaluation scores based on aggregated counts of correct decisions will be doubled for a system which structures this information as in DepBank. However, reproducing the exact DepBank **subord_form** relation from the GR **ccomp** one is non-trivial because DepBank treats modal auxiliaries as syntactic heads while the GR-scheme treats the main verb as head in all **ccomp** relations. We have not attempted to compensate for any further such discrepancies other than the one discussed in the previous paragraph. However, we do believe that they collectively damage scores for our system.

As King *et al.* note, it is difficult to identify such informational redundancies to avoid double-counting and to eradicate all system specific biases. However, reporting precision, recall and F_1-scores for each relation and feature separately and microaveraging these scores on the basis of a hierarchy, as in our GR scheme, ameliorates many of these problems and gives a better indication of the strengths and weaknesses of a particular parser, which may also be useful in a decision about its usefulness for a specific application. Unfortunately, Kaplan *et al.* do not report their results broken down by relation or feature so it is not possible, for example, on the basis of the arguments made above, to choose to compare the performance of our system on **ccomp** to theirs for **comp**, ignoring **subord_form**. King *et al.* do report individual results for selected features and relations from an evaluation of the complete XLE parser on all 700 DepBank sentences with an almost identical overall microaveraged F_1 score of 79.5%, suggesting that these results provide a reasonably accurate idea of the XLE parser's relative performance on different features and relations. Where we believe that the information captured by a DepBank feature or relation is roughly comparable to that expressed by a GR in our extended DepBank, we have included King *et al.*'s scores in the rightmost column in Table 1 for comparison purposes. Even if these features and relations were drawn from the same experiment, however, they would still not be *exactly* comparable. For instance, as discussed in §3 nearly half (just over 1K) the DepBank **subj** relations include **pro** as one element, mostly double counting a corresponding **xcomp** relation. On the other hand, our **ta** relation syntactically underspecifies many DepBank **adjunct** relations. Nevertheless, it is possible to see, for instance, that while both parsers perform badly on second objects ours is worse, presumably because of lack of lexical subcategorization information.

5 Conclusions

We have demonstrated that an unlexicalized parser with minimal manual modification for WSJ text – but no tuning of performance to optimize on this dataset alone, and no use of PTB – can achieve accuracy competitive with parsers employing lexicalized statistical models trained on PTB.

We speculate that we achieve these results because our system is engineered to make minimal use of lexical information both in the grammar and in parse ranking, because the grammar has been developed to constrain ambiguity despite this lack of lexical information, and because we can compute the full packed parse forest for all the test sentences efficiently (without sacrificing speed of processing with respect to other statistical parsers). These advantages appear to effectively offset the disadvantage of relying on a coarser, purely structural model for probabilistic parse selection. In future work, we hope to improve the accuracy of the system by adding lexical information to the statistical parse selection component without exploiting in-domain treebanks.

Clearly, more work is needed to enable more accurate, informative, objective and wider comparison of extant parsers. More recent PTB-based parsers show small improvements over Collins' Model 3 using PARSEVAL, while Clark and Curran (2004) and Miyao and Tsujii (2005) report 84% and 86.7% F_1-scores respectively for their own relational evaluations on section 23 of WSJ. However, it is impossible to meaningfully compare these results to those reported here. The reannotated DepBank potentially supports evaluations which score according to the degree of agreement between this and the original annotation and/or development of future consensual versions through collaborative reannotation by the research community. We have also highlighted difficulties for relational evaluation schemes and argued that presenting individual scores for (classes of) relations and features is both more informative and facilitates system comparisons.

6 References

Bikel, D.. 2004. Intricacies of Collins' parsing model, *Computational Linguistics*, 30(4):479–512.

Briscoe, E.J.. 2006. *An introduction to tag sequence grammars and the RASP system parser,* University of Cambridge, Computer Laboratory Technical Report 662.

Briscoe, E.J. and J. Carroll. 2002. Robust accurate statistical annotation of general text. In *Proceedings of the 3rd Int. Conf. on Language Resources and Evaluation (LREC)*, Las Palmas, Gran Canaria. 1499–1504.

Carroll, J. and E.J. Briscoe. 2002. High precision extraction of grammatical relations. In *Proceedings of the 19th Int. Conf. on Computational Linguistics (COLING)*, Taipei, Taiwan. 134–140.

Carroll, J., E. Briscoe and A. Sanfilippo. 1998. Parser evaluation: a survey and a new proposal. In *Proceedings of the 1st International Conference on Language Resources and Evaluation*, Granada, Spain. 447–454.

Clark, S. and J. Curran. 2004. The importance of supertagging for wide-coverage CCG parsing. In *Proceedings of the 20th International Conference on Computational Linguistics (COLING-04)*, Geneva, Switzerland. 282–288.

Collins, M.. 1999. *Head-driven Statistical Models for Natural Language Parsing*. PhD Dissertation, Computer and Information Science, University of Pennsylvania.

Gildea, D.. 2001. Corpus variation and parser performance. In *Proceedings of the Empirical Methods in Natural Language Processing (EMNLP'01)*, Pittsburgh, PA.

Inui, K., V. Sornlertlamvanich, H. Tanaka and T. Tokunaga. 1997. A new formalization of probabilistic GLR parsing. In *Proceedings of the 5th International Workshop on Parsing Technologies (IWPT'97)*, Boston, MA. 123–134.

Kaplan, R., S. Riezler, T. H. King, J. Maxwell III, A. Vasserman and R. Crouch. 2004. Speed and accuracy in shallow and deep stochastic parsing. In *Proceedings of the HLT Conference and the 4th Annual Meeting of the North American Chapter of the ACL (HLT-NAACL'04)*, Boston, MA.

Kiefer, B., H-U. Krieger, J. Carroll and R. Malouf. 1999. A bag of useful techniques for efficient and robust parsing. In *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics*, University of Maryland. 473–480.

King, T. H., R. Crouch, S. Riezler, M. Dalrymple and R. Kaplan. 2003. The PARC700 Dependency Bank. In *Proceedings of the 4th International Workshop on Linguistically Interpreted Corpora (LINC-03)*, Budapest, Hungary.

Klein, D. and C. Manning. 2003. Accurate unlexicalized parsing. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*, Sapporo, Japan. 423–430.

Lin, D.. 1998. Dependency-based evaluation of MINIPAR. In *Proceedings of the Workshop at LREC'98 on The Evaluation of Parsing Systems*, Granada, Spain.

Manning, C. and H. Schütze. 1999. *Foundations of Statistical Natural Language Processing*. MIT Press, Cambridge, MA.

Miyao, Y. and J. Tsujii. 2005. Probabilistic disambiguation models for wide-coverage HPSG parsing. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics*, Ann Arbor, MI. 83–90.

Nunberg, G.. 1990. *The Linguistics of Punctuation*. CSLI Lecture Notes 18, Stanford, CA.

Sampson, G.. 1995. *English for the Computer*. Oxford University Press, Oxford, UK.

Watson, R.. 2006. Part-of-speech tagging models for parsing. In *Proceedings of the 9th Conference of Computational Linguistics in the UK (CLUK'06)*, Open University, Milton Keynes.

Watson, R., J. Carroll and E.J. Briscoe. 2005. Efficient extraction of grammatical relations. In *Proceedings of the 9th Int. Workshop on Parsing Technologies (IWPT'05)*, Vancouver, Ca..