Chapter 1

HIGH PRECISION EXTRACTION OF GRAMMATICAL RELATIONS

John Carroll
Cognitive and Computing Sciences
University of Sussex
Falmer, Brighton
BN1 9QH, UK
johnca@cogs.susx.ac.uk
Ted Briscoe
Computer Laboratory
University of Cambridge
JJ Thomson Avenue
Cambridge CB3 0FD, UK
ejb@cl.cam.ac.uk

Abstract A parsing system returning analyses in the form of sets of grammatical relations can obtain high precision if it hypothesises a particular grammatical relation only when it is certain that the relation is correct. We operationalise this technique—in a statistical parser using a manually-developed wide-coverage grammar of English—by only returning relations that form part of all analyses licensed by the grammar. We observe an increase in precision from 75% to over 90% (at the cost of a reduction in recall) on a test corpus of naturally-occurring text.

Keywords: Statistical parsing, Grammatical relations, Text annotation

1. Introduction

Head-dependent grammatical relationships (possibly labelled with a relation type) have been advocated as a useful level of representation for grammatical structure in a number of different large-scale language-processing tasks. For instance, in recent work on statistical treebank grammar parsing (e.g. **?**) high levels of accuracy have been reached using lexicalised probabilistic models over head-dependent relations. **?** create dependency treebanks semi-automatically

in order to induce dependency-based statistical models for parse selection. **?**, **?** and others have evaluated the accuracy of both phrase structure and dependency parsers by matching head-dependent relations against 'gold standard' relations, rather than matching (labelled) phrase structure bracketings. Research on unsupervised acquisition of lexical information from corpora, such as argument structure of predicates (**?**; **?**), word classes for disambiguation (**?**), and collocations (**?**), has used grammatical or head-dependent relations. Such relations also constitute a convenient intermediate representation in applications such as information extraction (**?**; **?**), and document retrieval on the Web (**?**).

A variety of different approaches have been taken for robust extraction of relations from unrestricted text. Dependency parsing is a natural technique to use, and there has been some work in that area on robust analysis and disambiguation (e.g. **?**; **?**). Finite-state approaches (e.g. **?**; Aït-Mokhtar, S. and J-P. Chanod, 1997; **?**) have used hand-coded transducers to recognise linear configurations of words and to assign labels to words associated with, for example, subject/object-verb relationships. An intermediate step may be to mark nominal, verbal etc. 'chunks' in the text and to identify the head word of each of the chunks. Statistical finite-state approaches have also been used: **?** train a cascade of hidden Markov models to tag words with their grammatical functions. Approaches based on memory based learning have also used chunking as a first stage, before assigning grammatical relation labels to heads of chunks (**?**; **?**). **?** assume a richer input representation consisting of labelled trees produced by a treebank grammar parser, and use the treebank again to train a further procedure that assigns grammatical function tags to syntactic constituents in the trees. Alternatively, a hand-written grammar can be used that produces similar phrase structure analyses and perhaps partial analyses from which grammatical relations are extracted (e.g. **?**; **?**).

Recently, **?** have described an algorithm for computing expected *governor labels* for terminal words in labelled headed parse trees produced by a probabilistic context-free grammar. A governor label (implicitly) encodes a grammatical relation type (such as subject or object) and a governing lexical head. The labels are *expected* in the sense that each is weighted by the sum of the probabilities of the trees giving rise to it, and are computed efficiently by processing the entire parse forest rather than individual trees. The resulting set of governing-head tuples will not typically constitute a globally coherent analysis, but may be useful for interfacing to applications that primarily accumulate fragments of grammatical information from text (such as for instance information extraction, or systems that acquire lexical data from corpora). The approach is not so suitable for applications that need to interpret complete and consistent sentence structures (such as the analysis phase of transfer-based machine translation). Schmid and Rooth have implemented the algorithm for parsing with a lexicalised probabilistic context-free grammar of English and

applied it in an open domain question answering system, but they do not give any practical results or an evaluation.

In this paper we investigate empirically Schmid and Rooth's proposals, using a wide-coverage parsing system applied to a test corpus of naturally-occurring text, extend it with various thresholding techniques, and observe the trade-off between precision and recall in grammatical relations returned. Using the most conservative threshold results in a parser that returns only grammatical relations that form part of all analyses licensed by the grammar. In this case, precision rises to over 90%, as compared with a baseline of 75%.

2. The Analysis System

We extended an existing statistical shallow parsing system for English (e.g. **?**). Briefly, the system works as follows: input text is tokenised and then labelled with part-of-speech (PoS) tags by a tagger, and these are parsed using a wide-coverage unification-based grammar of English PoS tags and punctuation. For disambiguation, the parser uses a probabilistic LR model derived from parse tree structures in a treebank, augmented with a set of lexical entries for verbs, acquired automatically from a 10 million word sample of the British National Corpus (**?**), each entry containing subcategorisation frame information and an associated probability. The parser is therefore 'semi-lexicalised' in that verbal argument structure is disambiguated lexically, but the rest of the disambiguation is purely structural.

The coverage of the grammar—the proportion of sentences for which at least one complete spanning analysis is found—is around 80% when applied to the SUSANNE corpus (**?**). In addition, the system is able to perform parse failure recovery, finding the highest scoring sequence of phrasal fragments (following the approach of **?**), and the system has produced at least partial analyses for over 98% of the sentences in the 90M word written part of the British National Corpus.

The parsing system reads off grammatical relation tuples (GRs) from the constituent structure tree that is returned from the disambiguation phase. Information is used about which grammar rules introduce subjects, complements, and modifiers, and which daughter(s) is/are the head(s), and which the dependents. In the evaluation reported in **?**, the system achieves GR accuracy that is comparable to published results for other systems: extraction of non-clausal subject relations with 83% precision, compared with figure of 80% (**?**); and overall F_1-score[1] of unlabelled head-dependent pairs of 80%, as opposed to 83% (**?**)[2] and 84% (**?**—this with respect only to binary relations, and omitting the analysis of control relationships). **?** report an F_1-score of 87% for assigning grammatical function tags to constituents, but the task, and therefore the scoring method, is rather different.

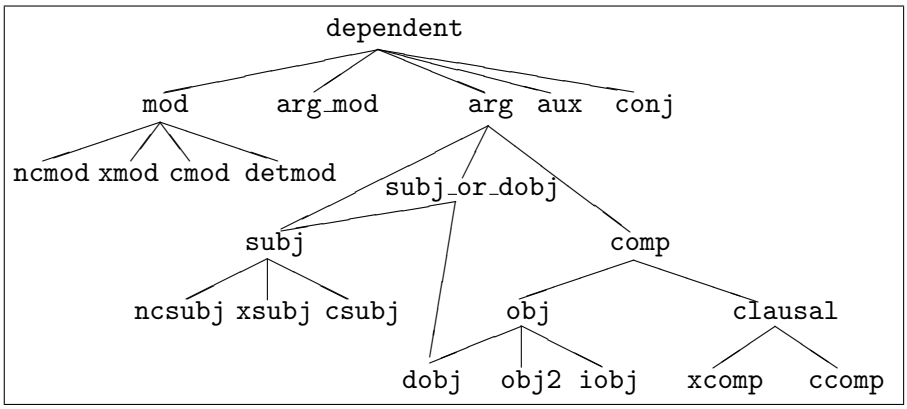

Figure 1.1. Grammatical relation hierarchy.

For the work reported in this paper we have extended the basic system, implementing a version of Schmid and Rooth's expected governor technique (see section 1 above) but adapted for unification-based grammar and GR-based analyses. Each sentence is analysed as a set of weighted GRs where the weight associated with each grammatical relation is computed as the sum of the probabilities of the parses that relation was derived from, divided by the sum of the probabilities of all parses. So, if we assume that Schmid and Rooth's example sentence *Peter reads every paper on markup* has 2 parses, one where *on markup* attaches to the preceding noun having overall probability 0.007 and the other where it has verbal attachment with probability 0.003, then some of the weighted GRs would be:

1.0 ncsubj(reads, Peter, _)
0.7 ncmod(on, paper, markup)
0.3 ncmod(on, reads, markup)

The GR scheme is described in detail by **?**. Figure 1.1 gives the full set of named relation types represented as a subsumption hierarchy. The most generic relation between a head and a dependent is *dependent*. Where the relationship between the two is known more precisely, relations further down the hierarchy can be used, for example *mod*ifier or *arg*ument. Relations *mod*, *arg_mod*, *aux*, *clausal*, and their descendants have slots filled by a type, a head, and its dependent; *arg_mod* has an additional fourth slot initial_gr. Descendants of *subj*, and also *dobj* have the three slots head, dependent, and initial_gr. Relation conj has a type slot and one or more head slots. The *nc*, *x* and *c* prefixes to relation names differentiate non-clausal, clausal and externally-controlled clausal dependents, respectively. Figure 1.2 contains a more extended example of a weighted GR analysis for a short sentence from the SUSANNE corpus.

1.0	aux(_, continue, will)	0.4490	iobj(on, place, tax-payers)
1.0	detmod(_, burden, a)	0.3276	ncmod(on, burden, tax-payers)
1.0	dobj(do, this, _)	0.2138	ncmod(on, place, tax-payers)
1.0	dobj(place, burden, _)	0.0250	xmod(to, continue, place)
1.0	ncmod(_, burden, disproportionate)	0.0242	ncmod(_, Fulton, tax-payers)
1.0	ncsubj(continue, Failure, _)	0.0086	obj2(place, tax-payers)
1.0	ncsubj(place, Failure, _)	0.0086	ncmod(on, burden, Fulton)
1.0	xcomp(to, Failure, do)	0.0020	mod(_, continue, place)
0.9730	clausal(continue, place)	0.0010	ncmod(on, continue, tax-payers)
0.9673	ncmod(_, tax-payers, Fulton)		

Figure 1.2. Weighted GRs for the sentence *Failure to do this will continue to place a disproportionate burden on Fulton taxpayers.*

3. Empirical Results

3.1 Weight Thresholding

Our first experiment compared the accuracy of the parser when extracting GRs from the highest ranked analysis (the standard probabilistic parsing setup) against extracting weighted GRs from all parses in the forest. To measure accuracy we use the precision, recall and F_1-score measures of parser GRs against 'gold standard' GR annotations in a 10,000-word test corpus of in-coverage sentences derived from the SUSANNE corpus and covering a range of written genres[3]. GRs are, in general, compared using an equality test, except that in a specific, limited number of cases the parser is allowed to return *mod*, *subj* and *clausal* relations rather than the more specific ones they subsume, and to leave unspecified the filler for the type slot in the *mod*, *iobj* and *clausal* relations[4]. The head and dependent slot fillers are in all cases the base forms of a single (head) word.

When a parser GR has a weight of less than one, we proportionally discount its contribution to the precision and recall scores. Thus, given a set T of GRs with associated weights produced by the parser, i.e.

$$T = \{(w_i, t_i) \mid w_i \text{ is the weight associated with GR } t_i, \text{where } 0 < w_i \leq 1\}$$

and a set S of gold-standard (unweighted) GRs, we compute the weighted match between S and the elements of T as

$$m = \sum_{(w_i,t_i)\in T} w_i \, \delta(t_i \in S)$$

Table 1.1. GR accuracy comparing extraction from just the highest-ranked parse compared to weighted GR extraction from all parses.

	Precision (%)	*Recall (%)*	F_1*-score*
Best parse	76.25	76.77	76.51
All parses	74.63	75.33	74.98

where $\delta(x) = 1$ if x is true and 0 otherwise. The weighted precision and recall are then

$$\frac{m}{\sum_{(w_i,t_i)\in T} w_i} \quad \text{and} \quad \frac{m}{|S|}$$

respectively, expressed as percentages. We are not aware of any previous published work using weighted precision and recall measures, although there is an option for associating weights with complete parses in the distributed software implementing the PARSEVAL scheme (**?**) for evaluating parser accuracy with respect to phrase structure bracketings. The weighted measures make sense for application tasks that can utilise potentially incomplete sets of potentially mutually-inconsistent GRs.

In this initial experiment, precision and recall when extracting weighted GRs from all parses were both one and a half percentage points lower than when GRs were extracted from just the highest ranked analysis (see table 1.1)[5]. This decrease in accuracy might be expected, though, given that a true positive GR may be returned with weight less than one, and so will not receive full credit from the weighted precision and recall measures.

However, these results only tell part of the story. An application might only utilise GRs which the parser is fairly confident are correct. For instance, in unsupervised acquisition of lexical information (such as subcategorisation frames for verbs), the usual methodology is to (partially) analyse the text, retaining only reliable hypotheses which are then filtered based on the amount of evidence for them over the corpus as a whole. Thus, **?** only creates hypotheses on the basis of instances of verb frames that are reliably and unambiguously cued by closed class items (such as pronouns) so there can be no other attachment possibilities. In recent work on unsupervised learning of prepositional phrase disambiguation, **?** derive training instances only from relevant data appearing in syntactic contexts that are guaranteed to be unambiguous. In our system, the weights on GRs indicate how certain the parser is of the associated relations being correct. We therefore investigated whether more highly weighted GRs are in fact more likely to be correct than ones with lower weights. We did this by setting a *threshold* on the output, such that any GR with weight lower than the threshold is discarded.

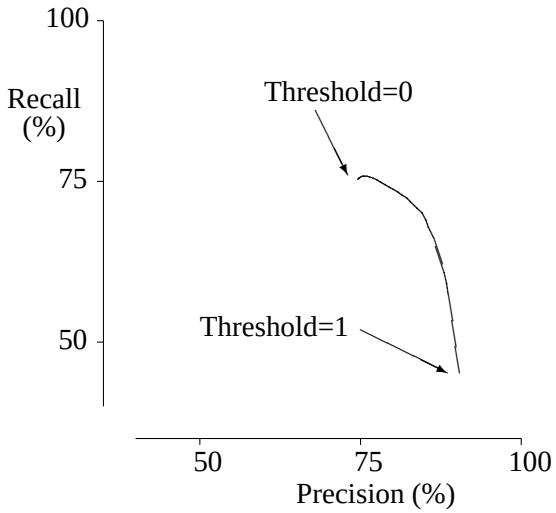

Figure 1.3. Weighted GR accuracy as the threshold is varied.

Figure 1.3 plots weighted recall and precision as the threshold is varied between zero and one The results are intriguing. Precision increases monotonically from 74.6% at a threshold of zero (the situation as in the previous experiment where all GRs extracted from all parses in the forest are returned) to 90.4% at a threshold of one. (The latter threshold has the effect of allowing only those GRs that form part of every single analysis to be returned). The influence of the threshold on recall is equally dramatic, although, since we have not escaped the usual trade-off with precision, the results are somewhat less positive. Recall decreases from 75.3% to 45.2%, initially rising slightly, then falling at a gradually increasing rate. At about the same point, precision shows a sharp rise, although smaller in magnitude. Table 1.2 shows in detail what is happening in this region. Between thresholds 0.99 and 1.0 there is only a two percentage point difference in precision, but recall differs by almost fourteen percentage points[6]. Over the whole range, as the threshold is increased from zero, precision rises faster than recall falls until the threshold reaches 0.65; here the F_1-score attains its overall maximum of 77%.

It turns out that the eventual figure of over 90% precision is not due to 'easier' GR types (such as the that between a determiner and a noun) being returned and more difficult ones (for example, that between a verb and a clausal complement) being ignored. Table 1.3 shows that the the majority of relation types are produced with frequency consistent with the overall 45% recall figure. Exceptions are *arg_mod* (encoding the English passive 'by-phrase') and *iobj* (indirect object), for which no GRs at all are produced. The reason for this is that both types of relation originate from an occurrence of a prepositional phrase

Table 1.2. Weighted GR accuracy as the threshold approaches 1.

GR Weight Threshold	*Precision* (%)	*Recall* (%)	F_1-*score*
1.0	90.40	45.21	60.27
0.99999999	90.27	46.28	61.19
0.9999999	90.17	46.87	61.68
0.999999	90.08	47.64	62.32
0.99999	90.03	48.91	63.38
0.9999	89.68	51.15	65.15
0.999	89.11	54.06	67.29
0.99	88.43	59.13	70.87
0.9	86.39	66.27	75.00
⋮	⋮	⋮	⋮
0.0	74.63	75.33	74.98

Table 1.3. Total numbers of parser and test corpus GRs by type, using a threshold of 1.

Relation Type	*Parser GRs*	*Test Corpus GRs*
mod	1915	2710
ncmod	979	2377
xmod	14	170
cmod	51	163
detmod	840	1124
arg_mod	0	39
arg	1058	1941
subj	664	993
ncsubj	659	984
xsubj	0	5
csubj	2	4
subj_or_dobj	852	1339
comp	394	948
obj	205	559
dobj	188	396
obj2	17	19
iobj	0	144
clausal	189	389
xcomp	161	323
ccomp	26	66
aux	237	379
conj	60	164

in contexts where it could be either a modifier or a complement of a predicate. This pervasive ambiguity means that there will always be disagreement between analyses over the relation type (but not necessarily over the identity of the head and dependent themselves).

3.2 Parse Unpacking

Schmid and Rooth's algorithm computes expected governors efficiently by using dynamic programming and processing the entire parse forest rather than individual trees. In contrast, we unpack the whole parse forest and then extract weighted GRs from each tree individually. Our implementation is certainly less elegant, but in practical terms for sentences where there are relatively small numbers of parses the speed is still acceptable. However, throughput goes down linearly with the number of parses, and when there are many thousands of parses—and particularly also when the sentence is long and so each tree is large—the system becomes unacceptably slow.

One possibility to improve the situation would be to extract GRs directly from forests. At first glance this looks viable: although our parse forests are produced by a probabilistic LR parser using a unification-based grammar, they are similar in content to those computed by a probabilistic context-free grammar, as assumed by Schmid and Rooth's algorithm. However, there are problems. If the test for being able to pack local ambiguities in the unification grammar parse forest is feature structure subsumption, unpacking a parse apparently encoded in the forest can fail due to non-local inconsistency in feature values (**?**)[7], so every GR hypothesis would have to be checked to ensure that the parse it came from was globally valid. It is likely that this verification step would cancel out the efficiency gained from using an algorithm based on dynamic programming. This problem could be side-stepped (but at the cost of less compact parse forests) by instead testing for feature structure equivalence rather than subsumption. A second, more serious problem is that some of our relation types encode more information than is present in a single governing-head tuple (the non-clausal subject relation, for instance, encoding whether the surface subject is the 'deep' object in a passive construction); this information can again be less local and violate the conditions required for the dynamic programming approach.

Another possibility is to compute only the n highest ranked parses and extract weighted GRs from just those. The basic case where $n = 1$ is equivalent to the standard approach of computing GRs from the highest probability parse. Table 1.4 shows the effect on accuracy as n is increased in stages to 1000, using a threshold for GR extraction of 1; also shown is the previous setup (labelled 'unlimited') in which all parses in the forest are considered.[8] (All differences in precision in the table are significant to at least the 95% level, except between 1000 parses and an unlimited number). The results demonstrate that limiting

Table 1.4. Weighted GR accuracy using a threshold of 1, with respect to the maximum number of ranked parses considered.

Maximum Parses	*Precision* (%)	*Recall* (%)	F_1*-score*
1	76.25	76.77	76.51
2	80.15	73.30	76.57
5	84.94	67.03	74.93
10	86.73	62.47	72.63
100	89.59	51.45	65.36
1000	90.24	46.08	61.00
unlimited	90.40	45.21	60.27

processing to a relatively small, fixed number of parses—even as low as 100—comes within a small margin of the accuracy achieved using the full parse forest. These results are striking, in view of the fact that the grammar assigns more than 300 parses to over a third of the sentences in the test corpus, and more than 1000 parses to a fifth of them. Another interesting observation is that the relationship between precision and recall is very close to that seen when the threshold is varied (as in the previous section); there appears to be no loss in recall at a given level of precision. We therefore feel confident in unpacking a limited number of parses from the forest and extracting weighted GRs from them, rather than trying to process all parses. We have tentatively set the limit to be 1000, as a reasonable compromise in our system between throughput and accuracy.

3.3 Parse Weighting

The way in which the GR weighting is carried out does not matter when the weight threshold is equal to 1 (since then only GRs that are part of every analysis are returned, each with a weight of 1). However, we wanted to see whether the precise method for assigning weights to GRs has an effect on accuracy, and if so, to what extent. We therefore tried an alternative approach where each GR receives a contribution of 1 from every parse, no matter what the probability of the parse is, normalising in this case by the number of parses considered. This tends to increase the numbers of GRs returned for any given threshold, so when comparing the two methods we found thresholds such that each method obtained the same precision figure (of roughly 83.38%). We then compared the recall figures (see table 1.5). The recall for the probabilistic weighting scheme is 4% higher (statistically significant at the 99.95% level) as expected, given the loss of information entailed by ignoring parse probabilities.

Table 1.5. Accuracy at the same level of precision using different weighting methods, with a 1000-parse tree limit.

Weighting Method	*Precision* (%)	*Recall* (%)	F_1*-score*
Probabilistic (at threshold 0.99)	88.38	59.19	70.90
Equally (at threshold 0.768)	88.39	55.17	67.94

It could be that an application has a preference for GRs that arise from less ambiguous sentences. In this case the parser could re-weight GRs such that the new weight is proportional to the inverse of the number of parses for the sentence: for instance changing weight w to

$$\left(\frac{1}{|P|}\right)^{(w-1)^2}$$

where $|P|$ is the number of parses. A weight of 1 would then be retained; however with this formula most values end up being either within a small region of 1, or extremely small. Using the absolute value of $w-1$ instead of $(w-1)^2$ seems to improve matters, but, in general, the best re-weighting method is likely to be application-specific and can only be determined empirically.

3.4 Maximal Consistent Relation Sets

It is interesting to see what happens if we compute for each sentence the maximal consistent set of weighted GRs. We might want to do this if we want complete and coherent sentence analyses, interpreting the weights as confidence measures over sub-analysis segments. We use a 'greedy' approximation to compute consistent relation sets, taking GRs sorted in order of decreasing weight and adding a GR to the set if and only if there is not already a GR in the set with the same dependent. (But note that the correct analysis may in fact contain more than one GR with the same dependent, such as the *ncsubj ... Failure* GRs in Figure 1.2, and in these cases this method will introduce errors). The weighted precision, recall and F_1-score at threshold zero are 79.31%, 73.56% and 76.33 respectively. Precision and F_1-score are significantly better (at the 95.95% level) than the baseline of all parses in table 1.1. Improvement in the algorithm used to compute consistent sets of GRs should increase this margin. This technique provides a way of building a complete analysis in terms of GRs which do not necessarily derive from a single syntactic phrase structure tree.

4. Conclusions and Further Work

We have extended a parsing system for English that returns analyses in the form of sets of grammatical relations, reporting an investigation into the extraction of *weighted* relations from probabilistic parses. We observed that setting a threshold on the output, such that any relation with weight lower than the threshold is discarded, allows a trade-off to be made between recall and precision. We found that by setting the threshold at 1 the precision of the system was boosted dramatically – from a baseline of 75% to over 90%. With this setting, the system returns only relations that form part of all analyses licensed by the grammar: the system can have no greater certainty that these relations are correct, given the knowledge that is available to it.

The technique is most appropriate for applications where a complete and consistent analysis is not required. However, the preliminary experiment reported in section 3.4 suggests that it can be extended to yield a high confidence consistent set of relations drawn from the set of *n*-best phrase structure analyses. Although we believe this technique to be especially well suited to statistical parsers, it could also potentially benefit any parsing system that can represent ambiguity and return analyses that are composed of a collection of elementary units. Such a system need not necessarily be statistical, since parse probabilities are not required when checking that a given sub-analysis segment forms part of all possible global analyses. Moreover, a statistical parsing system could use the technique to construct a reliable partially-annotated corpus automatically, which it could then be trained on.

One of our primary research goals is to explore unsupervised acquisition of lexical knowledge. The parser we use in this work is 'semi-lexicalised', using subcategorisation probabilities for verbs acquired automatically from (unlexicalised) parses. In the future we intend to acquire other types of lexico-statistical information (for example on PP attachment) which we will feed back into the parser's disambiguation procedure, bootstrapping successively more accurate versions of the parsing system. There is still plenty of scope for improvement in accuracy, since compared with the number of correct GRs in top-ranked parses there are roughly a further 20% that are correct but present only in lower-ranked parses. Table **??** gives the actual figures, broken down by relation type. There appears to be less room for improvement with argument relations (*ncsubj*, *dobj* etc.) than with modifier relations (*ncmod* and similar). This indicates that our next efforts should be directed to collecting information on modification.

Acknowledgments

We are grateful to Mats Rooth for early discussions about the expected governor label work. This research was supported by UK EPSRC projects

Relation Type	*In Parse Ranked 1*	*Not in Parse Ranked 1 but in Parses 2–1000*
ncmod	1691	538
xmod	56	36
cmod	99	65
detmod	1026	31
arg_mod	20	6
ncsubj	872	54
xsubj	4	1
csubj	1	1
dobj	337	31
obj2	16	1
iobj	109	34
xcomp	270	36
ccomp	65	6
aux	330	21
conj	114	24
total	5010	885

Table 1.6. Number of correct GRs in top-ranked parse, and number not in top-ranked parse but in others.

GR/N36462/93 ‘Robust Accurate Statistical Parsing (RASP)’ and by EU FP5 project IST-2001-34460 ‘MEANING: Developing Multilingual Web-scale Language Technologies’.

Notes

1. The F_1-score is defined as $2 \times precision \times recall/(precision + recall)$.

2. Our calculation is based on table 2 of **?**.

3. The annotated test corpus is available from http://www.cogs.susx.ac.uk/lab/nlp/carroll/greval.html.

4. We are currently refining the implementation of the extraction of GRs from parse trees, and will soon be able to remove these minor relaxations.

5. Ignoring the weights on GRs, standard (unweighted) evaluation results for all parses are: precision 36.65%, recall 89.42% and F_1-score 51.99.

6. Roughly, each percentage point increase or decrease in precision and recall is statistically significant at the 95% level. In this and all significance tests reported in this paper we use a one-tailed paired *t-test* (with 499 degrees of freedom).

7. The forest therefore also ‘leaks’ probability mass since it contains derivations that are in fact not legal.

8. At $n = 1000$ parses, the (unlabelled) weighted precision of head-dependent pairs is 91.0%.

References

Aït-Mokhtar, S. and J-P. Chanod (1997) Subject and object dependency extraction using finite-state transducers. In *Proceedings of the ACL/EACL Workshop on Automatic Information Extraction and Building of Lexical Semantic Resources*, 71–77. Madrid, Spain.

Argamon, S., I. Dagan and Y. Krymolowski (1998) A memory-based approach to learning shallow natural language patterns. In *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics*, 67–73. Montreal.

Blaheta, D. and E. Charniak (2000) Assigning function tags to parsed text. In *Proceedings of the 1st Conference of the North American Chapter of the Association for Computational Linguistics*, 234–240. Seattle, WA.

Bouma, G., G. van Noord and R. Malouf (2001) Alpino: wide-coverage computational analysis of Dutch. *Computational Linguistics in the Netherlands 2000. Selected Papers from the 11th CLIN Meeting*.

Brants, T., W. Skut and B. Krenn (1997) Tagging grammatical functions. In *Proceedings of the 2nd Conference on Empirical Methods in Natural Language Processing*, 64–74. Providence, RI.

Brent, M. (1993) From grammar to lexicon: unsupervised learning of lexical syntax. *Computational Linguistics*, 19(3), 243–262.

Briscoe, E. and J. Carroll (1997) Automatic extraction of subcategorization from corpora. In *Proceedings of the 5th Association for Computational Linguistics Conference on Applied Natural Language Processing*, 356–363. Washington, DC.

Briscoe, E. and J. Carroll (2002) Robust Accurate Statistical Annotation of General Text. In *Proceedings of the 3rd International Conference on Language Resources and Evaluation*, Gran Canaria, Canary Islands. 1499–1504.

Buchholz, S., J. Veenstra and W. Daelemans (1999) Cascaded grammatical relation assignment. In *Proceedings of the Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora*, College Park, MD. 239–246.

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*, 447–454. Granada, Spain.

Carroll, J., G. Minnen and E. Briscoe (1998) Can subcategorisation probabilities help a statistical parser?. In *Proceedings of the 6th ACL/SIGDAT Workshop on Very Large Corpora*. Montreal, Canada.

Clark, S. and D. Weir (2001) Class-based probability estimation using a semantic hierarchy. In *Proceedings of the 2nd Conference of the North American Chapter of the Association for Computational Linguistics*. Pittsburgh, PA.

Collins, M. (1999) *Head-driven statistical models for natural language parsing*. PhD thesis, University of Pennsylvania.

Grefenstette, G. (1997) SQLET: Short query linguistic expansion techniques, palliating one-word queries by providing intermediate structure to text. In *Proceedings of the RIAO'97*, 500–509. Montreal, Canada.

Grefenstette, G. (1998) Light parsing as finite-state filtering. In A. Kornai (Eds.), *Extended Finite State Models of Language*. Cambridge University Press.

Harrison, P., S. Abney, E. Black, D. Flickinger, C. Gdaniec, R. Grishman, D. Hindle, B. Ingria, M. Marcus, B. Santorini, & T. Strzalkowski (1991) Evaluating syntax performance of parser/grammars of English. In *Proceedings of the ACL Workshop on Evaluating Natural Language Processing Systems*, 71–78. Berkeley, CA.

Karlsson, F., A. Voutilainen, J. Heikkilä and A. Anttila (1995) *Constraint Grammar: a Language-Independent System for Parsing Unrestricted Text*. Berlin, Germany: de Gruyter.

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*, 473–480. University of Maryland.

Lafferty, J., D. Sleator and D. Temperley (1992) Grammatical trigrams: A probabilistic model of link grammar. In *Proceedings of the AAAI Fall Symposium on Probabilistic Approaches to Natural Language*, 89–97. Cambridge, MA.

Leech, G. (1992) 100 million words of English: the British National Corpus. *Language Research*, 28(1), 1–13.

Lin, D. (1998) Dependency-based evaluation of MINIPAR. In *Proceedings of the The Evaluation of Parsing Systems: Workshop at the 1st International Conference on Language Resources and Evaluation*. Granada, Spain (also available as University of Sussex technical report CSRP-489).

Lin, D. (1999) Automatic identification of non-compositional phrases. In *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics*, 317–324. College Park, MD.

McCarthy, D. (2000) Using semantic preferences to identify verbal participation in role switching alternations. In *Proceedings of the 1st Conference of the*

North American Chapter of the Association for Computational Linguistics, 256–263. Seattle, WA.

Oepen, S. and J. Carroll (2000) Ambiguity packing in constraint-based parsing — practical results. In *Proceedings of the 1st Conference of the North American Chapter of the Association for Computational Linguistics*, 162–169. Seattle, WA.

Palmer, M., R. Passonneau, C. Weir and T. Finin (1993) The KERNEL text understanding system. *Artificial Intelligence*, 63, 17–68.

Pantel, P. and D. Lin (2000) An unsupervised approach to prepositional phrase attachment using contextually similar words. In *Proceedings of the 38th Annual Meeting of the Association for Computational Linguistics*, 101–108. Hong Kong.

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

Schmid, H. and M. Rooth (2001) Parse forest computation of expected governors. In *Proceedings of the 39th Annual Meeting of the Association for Computational Linguistics*, 458–465. Toulouse, France.

Srinivas, B. (2000) A lightweight dependency analyzer for partial parsing. *Natural Language Engineering*, 6(2), 113–138.

Yeh, A. (2000) Using existing systems to supplement small amounts of annotated grammatical relations training data. In *Proceedings of the 38th Annual Meeting of the Association for Computational Linguistics*, 126–132. Hong Kong.