Relational Evaluation Schemes

Ted Briscoe*, **John Carroll**†, **Jonathan Graham***, **Ann Copestake***

*Computer Laboratory
University of Cambridge
{Ted.Briscoe, Ann.Copestake}@cl.cam.ac.uk

†Cognitive and Computing Sciences
University of Sussex
John.Carroll@cogs.susx.ac.uk

Abstract

We describe extensions to a scheme for evaluating parse selection accuracy based on named grammatical relations between lemmatised lexical heads. The scheme is intended to directly reflect the task of recovering grammatical and logical relations, rather than more arbitrary details of tree topology. There is a manually annotated test suite of 500 sentences which has been used by several groups to perform evaluations. We are developing software to create larger test suites automatically from existing treebanks. We are considering alternative relational annotations which draw a clearer distinction between grammatical and logical relations in order to overcome limitations of the current proposal.

1. Introduction

We have developed a scheme for evaluating parse selection accuracy based on named grammatical relations between lemmatised lexical heads. The scheme is intended to directly reflect the task of recovering semantic relations, rather than more arbitrary details of tree topology—as with the PARSEVAL scheme, which has been criticised frequently for the opaque relationship between its measures and such relations (Carroll *et al.*, 1998; Magerman, 1995; Srinivas, 1997). Carroll *et al.* (1998) provide more detailed motivation and comparison with other extant schemes.

Carroll *et al.* (1999, 2002 in press) report the development of a test suite of 500 sentences annotated with grammatical relations, the specification of the relations, and their criteria of application. The set of named relations are organised as a subsumption hierarchy in which, for example, subj(ect) underspecifies n(on)c(lausal)subj(ect). There are a total of 15 fully specified relations, however, many of these can be further subclassified; for example, subj relations have an initial-gr slot used to encode whether the syntactic subject is logical object (as in passive) and for other marked subjects (such as in locative inversion). Thus a fully specified GR might look like (ncsubj marry couple obj) to encode the subj relation in *The couple were married in August*, and the GR annotation of each sentence of the test suite consists of a set of GR *n*-tuples. Figure 1 gives the full set of named relations 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 x and c prefixes to relation names differentiate clausal control alternatives.

When the proprietor dies, the establishment should become a corporation until it is either acquired by another proprietor or the government decides to drop it.

```
(ncsubj die proprietor _)
(ncsubj become establishment _)
(xcomp _ become corporation)
(ncsubj acquire it obj)
(arg_mod by acquire proprietor subj)
(ncmod _ acquire either)
(ncsubj decide government _)
(xcomp to decide drop)
(ncsubj drop government _)
(dobj drop it _)
(cmod when become die)
(cmod until become acquire)
(cmod until become decide)
(detmod _ proprietor the)
(detmod _ establishment the)
(detmod _ corporation a)
(detmod _ proprietor another)
(detmod _ government the)
(aux _ become shall)
(aux _ acquire be)
(conj or acquire decide)
```

Figure 2: Grammatical relation sample annotation.

Figure 2 shows the GR encoding of a sentence from the Susanne corpus.

The evaluation metric uses the standard precision and recall and F_α measures over sets of such GRs. Carroll and Briscoe (2001) also make use of weighted recall and precision (as implemented in the PARSEVAL software) to evaluate systems capable of returning *n*-best sets of weighted GRs. The software makes provision for both averaged scores over all relations as well as scores by named relation. It also supports partial scoring in terms of non-leaf named relations which underspecify leaf relations. The current specification of the

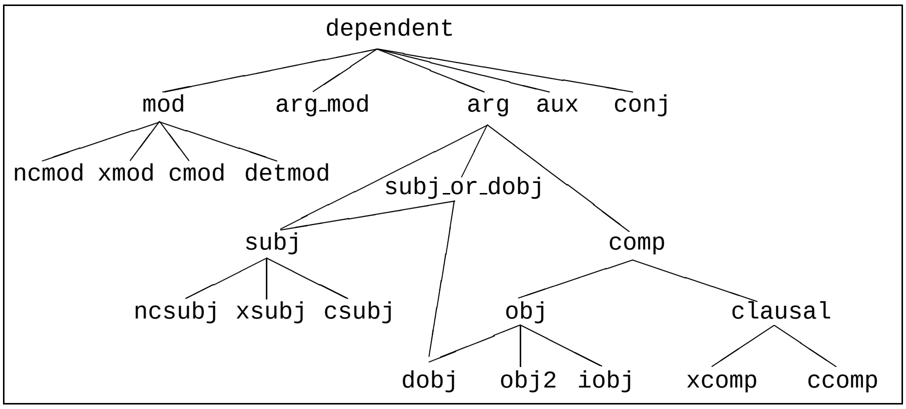

Figure 1: Grammatical relation hierarchy.

scheme along with the test suite and evaluation software (implemented in Common Lisp) is available from http://www.cogs.susx.ac.uk/lab/nlp/carroll/greval.html

Evaluation of stochastic parsers using relational schemes similar to our proposal is becoming more common (e.g. Collins, 1999; Lin, 1998; Srinivas, 2000). However, comparison across such results is hampered by the fact that the set of relations extracted is not standardised across these schemes, and it is clear that some relations (e.g. that between determiners and head nouns) are much easier to extract than others (e.g. control relations in predicative complements), as can be seen, for example, from the separate and divergent precision / recall results by named relation reported by Carroll *et al.* (1999). This makes meaningful comparison of 'headline results' such as mean overall F_1 measures very hard. Our scheme attempts to ameliorate these problems by supporting different levels of granularity within named relations (ncsubj / csub / xsubj $\subset$ subj) and encouraging not only the reporting of overall mean precision / recall scores, but also separate scores for each named relation.

In the rest of this paper we describe ongoing efforts to improve the evaluation scheme and enlarge the annotated test suite(s).

2. Divergent system output representations

There remain several infelicities in the current scheme that are a consequence of the method of factoring information into distinct relations which, in fact, still encode composites of information. For example, a system which clearly separates categorial constituency and functional information, such as one based on LFG, might choose to map F-structure SUBJ relations to subj in our scheme. A more constituency based parser might map NPs immediately dominated by S and preceding a VP to ncsubj, and Ss in the same configuration to csubj. Superficially the latter system is extracting more information because the relation name encodes categorial as well as relational information. The current scoring metric also assigns a penalty to systems that do not recover fully-specified (leaf) relations. However, for either system to score in the evaluation the subj relation most hold between lemmatised heads of the appropriate type, so the distinction between clausal and non-clausal subjects is maintained in both, since clausal subjects have verbal heads. On the other hand a system which systematically returned subj-or-dobj relations, as opposed to a leaf subj or obj one, would clearly be losing significant information pertinent to recovery of underlying logical relations.

There are many other cases of divergent encoding of aspects of categorial and functional information: for example, a LFG system will clearly distinguish clausal and predicative complements at F-structure corresponding directly to the xcomp / ccomp distinction in our relational scheme. However, a parser that represents such complements as clauses (S nodes) with or without an empty (PRO) NP subject, as in the Penn WSJ Treebank, would need to utilise a more complex (non-local) mapping from tree topology and node labels to named relations in order to maintain the xcomp / ccomp distinction. However, in this case, the easier underspecification to comp is genuinely significant since in either case the relation will hold between the same lexical (verbal) heads.

There are, in principle, two ways of dealing with such divergences. The first is to complicate the mapping from system output to named relations so that the specific set of leaf relations identified in the current scheme is recovered, if it is deducible from the total system output. The second is to modify the scoring metric so that informationally insignificant underspecification is not penalised. In some cases, such as the LFG system SUBJ case described above, the latter step will be much easier. In the new version of the specification and evaluation measure, we will attempt to identify such cases and parameterise the evaluation software to compute scores appropriately, as well as provide more specific guidance on mapping of named relations to the output of extant systems. This should improve the validity of cross-system evaluation. However, problems of this type are likely to emerge for each new system representation considered, so this is likely to be an ongoing process requiring judgement on the part of evaluators coupled with explicit description of decisions made alongside reported socres.

Provision of a flexible software system for mapping from parser output representations to factored relational ones may also ameliorate this class of problems (see section 5.). In particular, where a specific choice of system output representation necessitates a more complex mapping to leaf relations in our scheme, it would facilitate fair and feasible cross-system comparison if the evaluation scheme provided software that would recover the named leaf relations from the system output. Once again, each new system representation is likely to throw up new problems of this type, so flexible and easily parameterisable software will be more useful.

3. Surface / logical form divergence

The current annotation scheme attempts to stay close to surface grammatical structure, while also encoding divergence from predicate-argument structure / logical form. Divergence is currently encoded using two distinct mechanisms for different types of cases. Extra slots in named relations are used to indicate surface / underlying logical relation divergences, as with subj discussed in section 1. An additional relation is used for coordination (conj) to indicate how the conjunction scopes over the individual conjuncts.

One conspicuous area where the current scheme is inadequate is with equative and comparative constructions, which occur quite frequently in the 500 sentence test suite. Semantically, it is standard to treat *more* and *as*, etc as generalised quantifiers over propositions so that an example like

> *GR evaluation is more / as attractive than / as PARSEVAL*

is represented (very crudely) as

> more′(is-attr′(GReval′), is-attr′(PARSEVAL′))

This example, however, is annotated by the GRs

> (ncmod _ attractive more)
> (ncmod than attractive PARSEVAL)

However, in general, the GR annotation of such constructions is variable because of the varied surface syntactic location of *more* and *as* and also because of the optionality of and degree of ellipsis in the *than* / *as* constituent. Furthermore, because of the divergence between surface form and logical form the current annotations give little indication of whether a system would be capable of outputting an appropriate logical form. Replacing the current annotation with one close to the target logical form would undermine the scheme, since most extant stochastic parsers would be unable to generate such a representation.

One alternative is to additionally annotate such constructions with construction-specific named relations. This could be based on the approach to coordination, where the named relation

> (conj conj-type conjunct-heads+)

is used in addition to distributing the conjunct heads over multiple occurrences of the relation over the coordinate construction. For comparatives and equatives, we could add a relation like

> (compequ as/more/... attractive GReval PARSEVAL)

encoding the type of comparison, the predicate of comparison, and the arguments to this predicate.

There are undoubtedly further constructions, beyond coordination and comparatives / equatives that merit some such treatment. The advantage of adding additional construction-specific named relations that encode the same phenomena from different perspectives is that the resulting annotation will support a graded and fine-grained evaluation of the extent to which a specific system can support recovery of underlying logical form / predicate-argument structure in addition to surface grammatical relations. The disadvantage of this approach is that the scheme is likely to become more complex, and thus its recovery from any specific parser representation more time-consuming. In addition, the encoding of the underlying logical relations in the GR scheme has already spawned two divergent mechanisms, and may well require more.

4. MRS-style annotation scheme

A second and more complex but potentially more thorough approach to the issue of surface / logical form divergence is to bleach the current GR scheme of all attempts to represent such mismatches and instead define a factored and underspecified semantic annotation scheme to be used in tandem with GR annotation. The approach to underspecified logical representation developed by Copestake *et al.* (2001) can be extended to allow semantics to be underspecified to a much greater degree. In this extension of minimal recursion semantics (MRS), a Parsons-style notation (Parsons, 1990) is used, with explicit equalities representing variable bindings. For instance, from

> *The couple were married.*

a particular parsing system might return

> (ARGN u1 u2)
> (marry u3)
> (couple u4)

However, the fully specified test suite annotation would be

> (ARG2 e1 x4)
> (marry e2)
> (couple x3)
> e1 = e2
> x3 = x4

where ARG2 is formally a specialisation of ARGN, and the equalities and variable sorts also add information.

Potentially, this would allow us to dispense with complications like init-gr fields in the GR annotation and provide a principled basis for a graded evaluation of the recovery of logical form. The disadvantage over the further extension of the existing scheme is that two stages of extraction from specific system output are now required, the matching operations and scoring metrics become more complex, and the ability to do a graded evaluation of recovery of both grammatical and logical relations may be somewhat undermined.

```
try
  {
   while (dd)
    {
     String s = readWord(W);
     setS += 1;

     if (c==0) dd = false;

     if (s.equals("S"))
      {
       if (domprecedes("S", "NP",
                       "VP", setS))
        { String head = mainverb(setvp);
          String dependent =
            righthead("NP", "N-", setnp);
          String objslot =
            ispassive(setvp);
          System.out.println(
                   "(ncsubj " + head + "
                   " + dependent + "
                   " + objslot + ")");
        }
      }
    }
  }
```

Figure 3: The ncsubj extraction class.

5. Enlarging and improving the test suite(s)

The current test suite of 500 sentences is too small, but was still labour-intensive to create semi-automatically. Consequently, it contains a number of inadequacies: tokenisation of multiwords is somewhat arbitrary, some relations which should be included are systematically omitted (e.g. predicative XP complements of *be* have not been annotated with their controlled subjects), quotation marks have been systematically removed, and so forth. The next release will attempt to remove these inadequacies. However, it is clear that we also need a method for annotating much more data efficiently. To this end we have been developing a generic system, implemented in JAVA, that can be applied to existing treebanks to extract relational information (Graham, 2002). This system can, in principle, extract GRs in the current or related schemes, or even (possibly underspecified) MRSs. It can be parameterised for different extant treebanks, such as Penn Treebank-II or Susanne, and requires a set of declarative rules expressed in terms of tree topology and node labels for each named relation. The system has been designed to process labelled trees looking for relations defined ultimately in terms of (immediate) dominance and (immediate) precedence efficiently. It has been tested on a subset of GRs, concentrating particularly on the subj sub-hierarchy. A fragment of the class for ncsubj encoding relevant constraints is shown in Figure 3, giving a sense of the degree of parameterisation required for different representations. Running a first prototype of the GR extractor on the 30 million word automatically annotated WSJ BLLIP corpus distributed by the LDC results in estimated recovery of 86% of ncsubj and dobj relations with a precision of 84%, taking around 3 hours CPU time on standard hardware.

This system will facilitate rapid automatic construction of relational annotation according to specified input and output scheme(s) up to the limit of what is currently represented in treebanks and system output. Our longer term plan is to make this software, and a number of rule sets implemented in it, available as part of the evaluation scheme. This should facilitate both the construction of test data and the mapping of system output to the required format.

6. Conclusions

Relational schemes for parser evaluation are gaining in popularity over the exclusive use of PARSEVAL or similar tree topology based measures. We hope that the ongoing work reported here will facilitate further cross-system and within-system relational evaluation. To this end, we are developing test suites and software to support flexible mapping from system and treebank output to relational encodings of grammatical and underlying logical relations, and actively seeking feedback from the community on weaknesses of our current encoding scheme and evaluation measures and errors in our current test set.

Acknowledgements

We would like to thank Ron Kaplan for carefully documenting many errors and inconsistencies in our semi-automatic annotation of the 500 word test suite. The GR encoding scheme was heavily influenced by the EAGLES encoding scheme, primarily developed by Antonio Sanfilippo. We would also like to thank Anne Abeillé and Srinivas Bangalore for useful discussions. This work was partially supported by the EPSRC-funded RASP project (grants GR/N36462 and GR/N36493).

References

Carroll, J. and E. Briscoe (2001) ‘High precision extraction of grammatical relations’, *Proceedings of the 7th ACL/SIGPARSE International Workshop on Parsing Technologies (IWPT’01)*, Beijing, China, pp. 78–89.

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

Carroll, J., G. Minnen and E. Briscoe (1999) ‘Corpus annotation for parser evaluation’, *Proceedings of the EACL-99 Post-Conference Workshop on Linguistically Interpreted Corpora (LINC’99)*, Bergen, Norway, pp. 35–41.

Carroll, J., G. Minnen and E. Briscoe (2002, in press) ‘Parser evaluation using a grammatical relation annotation scheme’ in Abeille, A. (ed.), *Treebanks: Building and Using Syntactically Annotated Corpora*, Dordrecht: Kluwer.

Collins, M. (1999) *Head-driven Statistical Models for Natural Language Parsing*, PhD Dissertation, University of Pennsylvania.

Copestake, A., A. Lascarides and D. Flickinger (2001) ‘An algebra for semantic construction in constraint-based grammars’, *Proceedings of the 39th Annual Meeting of the Association for Computational Linguistics*, Toulouse, France, pp. 132–139.

Graham, J. (2002, in preparation) *From Treebank to Lexicon*, DPhil Dissertation, University of Cambridge, Computer Laboratory.

Lin, D. (1998) 'Dependency-based evaluation of MINIPAR', *Proceedings of the The Evaluation of Parsing Systems: Workshop at the 1st International Conference on Language resources and Evaluation*, Granada, Spain.

Magerman, D. (1995) *Natural Language Parsing as Statistical Pattern Recognition*, PhD Dissertation, Stanford University.

Parsons, T. (1990) *Events in the Semantics of English*, MIT Press, Cambridge, MA.

Srinivas, B. (1997) *Complexity of Lexical Descriptions and its Relevance to Partial Parsing*, PhD Dissertation, University of Pennsylvania.

Srinivas, B. (2000) 'A lightweight dependency analyzer', *Natural Language Engineering, vol.6.2*, 113–138.