Projecting RMRS from TIGER dependencies

Kathrin Spreyer
Saarland University and DFKI

Anette Frank
DFKI, Saarbrücken

Proceedings of the 12th International Conference on
Head-Driven Phrase Structure Grammar

Department of Informatics, University of Lisbon

Stefan Müller (Editor)

2005

Stanford, CA: CSLI Publications

pages 354–363

Spreyer, Kathrin & Anette Frank. 2005. Projecting RMRS from TIGER dependencies. In Stefan Müller (ed.), *Proceedings of the 12th International Conference on Head-Driven Phrase Structure Grammar, Department of Informatics, University of Lisbon*, 354–363. Stanford, CA: CSLI Publications. DOI: 10.21248/hpsg.2005.20.

Abstract

We present a method for automatic RMRS semantics construction from dependency structures, following the semantic algebra of Copestake et al. (2001). We have applied this method to a subset of the TIGER Dependency Bank for German (Forst et al., 2004) to obtain a semantic treebank for (HPSG) parser evaluation. We describe the semantics construction mechanism and give evaluation figures from manual validation of the treebank. These indicate high precision of the automatic RMRS construction process.

1 Introduction

Treebanks are under development for many languages. They are exploited for induction of treebank grammars, training of stochastic parsers, and for evaluating and benchmarking competitive parsing and grammar models. While parser evaluation against treebanks is most natural for treebank-derived grammars, it is extremely difficult for hand-crafted grammars that represent higher-level syntactic or semantic information, such as LFG, HPSG, or CCG grammars (cf. Carroll et al., 2002).

In a recent joint initiative, the TIGER project provides dependency-based treebank representations for German, on the basis of the TIGER treebank (Brants et al., 2002). Forst (2003) applied treebank conversion methods to the TIGER treebank, to derive an f-structure bank for stochastic training and evaluation of a German LFG parser. A more general, theory-neutral dependency representation is currently derived from this TIGER-LFG treebank, to enable cross-framework parser evaluation (Forst et al., 2004). However, while Penn-treebank style grammars and LFG analyses are relatively close to dependency representations (cf. Crouch et al., 2002; Kaplan et al., 2004), the situation is different for grammar formalisms that deliver deeper semantic representations, such as HPSG or CCG.

In order to provide a closer evaluation standard and appropriate training material for German HPSG grammars, we propose a method for semi-automatic construction of an RMRS treebank for German on the basis of the TIGER-Dependency Bank. In contrast to treebanks constructed from the analyses of hand-crafted grammars, this method achieves a gold standard for comparative parser evaluation where the upper bound for coverage is defined by the corpus (here, German newspaper text), not by the grammar.

Our treebank conversion method effectively implements RMRS semantics construction from dependency structures, and can be further developed to a general method for RMRS construction from LFG f-structures, similar to recent work in the LOGON project (Dyvik et al., 2005).

[†]The research reported here was conducted in cooperation with the TIGER project and has partially been supported by the project QUETAL (DFKI), funded by the German Ministry for Education and Research, grant no. 01 IW C02. Special thanks go to Dan Flickinger and Berthold Crysmann for advice on theoretical and grammar-specifi c issues. We also thank Martin Forst, who provided us with the TIGER DB dependency structures, Berthold Crysmann for providing HPSG lexical resources, and Ulrich Sch¨afer for XSLT-scripts used for visualisation.

2 The TIGER Dependency Bank

The input to the RMRS construction process consists of dependency representations of the TIGER Dependency Bank (TIGER-DB) (Forst et al., 2004). The TIGER-DB is derived from (a subset of) the TIGER treebank. It abstracts away from constituency in order to remain as theory-neutral as possible. The TIGER-DB was derived semi-automatically from the TIGER-LFG Bank of Forst (2003), by defining various normalisations. The dependency format is similar to the Parc 700 Dependency Bank (King et al., 2003). So-called dependency triples are sets of two-place predicates that encode grammatical relations. The arguments represent the head of the dependency and the dependent, respectively. The triples further retain a number of morpho-syntactic features from the LFG representations, such as agreement information for nominals and adjectives, or tense information. Figure 1 displays a sample dependency representation.

```
sb(müssen~0, Museum~1)
case(Museum~1, nom)
gend(Museum~1, neut)
num(Museum~1, sg)
mod(Museum~1, privat~1001)
cmpd_lemma(Museum~1, Privatmuseum)
oc_inf(müssen~0, weichen~3)
mood(müssen~0, ind)
tense(müssen~0, pres)
sb(weichen~3, Museum~1)
```

Figure 1: TIGER-DB structure for *Privatmuseum muss weichen* – Private museum deemed to vanish.

However, dependency structures are difficult to match against the output of HPSG parsing. HPSG analyses do not come with an explicit representation of functional structure, but directly encode semantic structures, in terms of MRS (Copestake et al., 2005) or RMRS (Copestake, 2003). This leaves a gap to be bridged in terms of normalisation of diathesis, the encoding of arguments vs. adjuncts, the representation of constructions like relative clauses, and the representation of quantifiers and their scoping relations.

In order to provide a gold standard that can be matched against the output of HPSG parsing for evaluation, and further, for training stochastic grammar models, we propose a method for treebank conversion that essentially performs RMRS construction from LFG-based dependency representations.

For the purpose of semantics construction, the triples format has both advantages and disadvantages. On the one hand, the LFG-derived dependencies offer all the advantages of a functional as opposed to a constituency-based representation. This representation already filters out the semantically inappropriate status of auxiliaries as heads; their contribution is encoded by features such as `perf` or `fut`, which can be directly translated into features of semantic event variables.

Most importantly, the triples localize dependencies which are not locally realized in terms of phrase structure (as e. g. in control structures, coordination, or long-distance constructions), so that when constructing the semantics from the dependency format, we do not need additional mechanisms to identify the arguments of a governing predicate.

The challenges we face mainly concern the lack of constituency information in the dependency representations. Yet, it is possible to reconstruct important phrase-structural information from the dependency input format (see Section 3.3. below).

3 RMRS Construction from TIGER Dependencies

3.1 Treebank Conversion by Term Rewriting

Similar to Forst (2003) we are using the term rewriting system of Crouch (2005) for treebank conversion. Originally designed for Machine Translation, the system is a powerful tool for structure rewriting that is also applied to other areas of NLP, such as the induction of knowledge representations (Crouch, 2005).

The input to the system consists of a set of facts in a prolog-like term representation. The rewrite rules refer to these facts in the left-hand side (LHS), either conjunctively (expressed by separating conjuncts with a comma '`,`') or disjunctively (expressed by '|'). Expressions on the LHS may be negated by a prefixed '`-`', thereby encoding negative constraints for matching. A rule applies if and only if all facts specified on the LHS are satisfied by the input set of facts. The right-hand side (RHS) of a rewrite rule defines a conjunction of facts which are added to the input set of facts if the rule applies. The system further allows the user to specify whether a matched fact will be consumed (i. e., removed from the set of facts) or whether it will be retained in the rule's output set of facts (marked by the prefix '`+`').

The processing of rules is *strictly ordered*. The rules are applied in the order of textual appearance. Each rule is tested against the current input set of facts and, if it matches, produces an output set of facts that provides the input for the next rule in sequence. Each rule applies concurrently to all distinct sets of matching facts, i.e. it performs parallel application in case of alternative matching facts.

The system offers powerful rule encoding facilities, in terms of macros and templates. These abstraction means help the user to define rules in a perspicious and modular way.

3.2 RMRS Construction

Within the formal framework of HPSG, every lexical item defines a complete RMRS structure. Semantics composition rules are defined in parallel with syntactic composition. In each composition step, the RMRSs of the daughters are combined according to strict semantic composition rules, to yield the RMRS representation of the phrase (cf. Copestake et al., 2001). Following the scaffolding of

the syntactic structure in this way finally yields the semantic representation of the sentence.

For our task, the input to semantics construction is a dependency structure. As established by work on Glue Semantics (Dalrymple, 1999), semantics construction from dependency structures can in similar ways proceed recursively, to deliver a semantic projection of the sentence. However, the resource-based construction mechanism of Glue Semantics leads to alternative derivations in case of scope ambiguities. In contrast to Glue, we target an underspecified semantic representation. Although usually defined on phrasal configurations, the algebra for (R)MRS construction as defined in Copestake et al. (2001) is neutral with regard to the syntactic representation, and can be transposed to composition on the basis of dependency relations, much alike the Glue framework.

Yet, the rewriting system we are using is not suited for a recursive application scheme: the rules are strictly ordered, and each rule simultaneously applies to all facts that satisfy the constraints. That is, the RMRS composition cannot recursively follow the composition of dependents in a given input structure.

The RMRS Skeleton. RMRS construction is thus designed around one *global RMRS*, featuring a TOP label, a RELS set containing the *elementary predications* (EPs), a set HCONS of *handle constraints* which state restrictions on possible scopes, and a set of ING constraints that represent the *in-group* relation.[1]

When defining composition, instead of projecting and accumulating RMRS constraints step-wise by recursive composition rules from the lexical items to the top level of the sentence, at each step we directly insert all EPS, ING and HCONS constraints into the global RMRS, i.e. the RMRS with the top handle. The semantics composition rules are thus reduced to the inherent semantic operations of the algebra of Copestake et al. (2001): the binding of argument variables and encoding of scope constraints. These basic semantic operations are defined by appropriate definitions and operations on the HOOK features in the composition rules.

Lexical RMRSs. The notion of *lexical RMRSs* as it is defined here slightly differs from the standard one. If semantic composition proceeds along a tree structure, lexical RMRSs are constructed at the leaf nodes. In our scenario, a lexical RMRS is projected from the PRED features in the dependency structures, irrespective of any arguments, which are considered by subsequent composition rules.

We define the lexical RMRSs in two steps: First, the hook label is (freely) instantiated and thus available for reference to this RMRS by other rules. Second, the hook variable and the basic semantics (EPs for the relation and the ARG0, at least) are introduced on the basis of the predicate's category. This category information is not explicit in the dependencies, but it can be induced from the

[1] Whenever two handles are related via an ing constraint, they can be understood to be conjoined. This is relevant, e.g., for intersective modification, since a quantifier that outscopes the modified noun must also take scope over the modifier.

(a)
```
add_ep(Lb,Type,Feat,Val) ::
     +rels(_,Rels)
==>  ep(Rels,EP), type(EP,Type),
     lb(EP,Lb), complex_term(Feat,EP,Val).
```

(b)
```
     +pred(X,Pred), -mo(_,X), -spec(_,X),
     +'s::'(X,SemX), +hook(SemX,Hook), +lb(Hook,Lb)
==>  var(Hook,Var)
&&   add_ep(Lb,ep_rel,rel,Pred)
&&   add_ep(Lb,ep_arg0,arg0,Var).
```

(c)
$$\begin{bmatrix} \textsc{Top} & \textit{handle} \\ \textsc{Rels} & \left\{ \ldots, \begin{bmatrix} \textit{\_Riese\_n} & \\ \textsc{Lb} & h \\ \textsc{Arg0} & x \end{bmatrix}, \ldots \right\} \\ \textsc{Hcons} & \{\ldots\} \\ \textsc{Ing} & \{\ldots\} \end{bmatrix}$$

Figure 2: (a) Expansion of `add_ep` template, (b) a rule with a template call, (c) the output lexical RMRS.

grammatical function borne by the predicate, as well as the presence or absence of certain morphological features.

Figure 2 shows a sample lexical RMRS and the rule that yields it: The rule in (b) applies to predicates, i.e. to `pred` features, with a value `Pred` and a hook label `Lb`. In the RHS, one EP is added for the relation represented by `Pred`, and one for the ARG0, which is identified with the hook variable.[2]

Composition. The semantic composition of arguments and functors makes use of a predicate `arg/3` which encodes the argument structure of the predicates.[3] Given a predicate `arg(Fctor,N,Arg)`, the binding of the argument to the functor is steered by the previously defined hooks of the two semantic entities in that the matching rule attaches an EP with an attribute ARG*N* to the externalized label in the functor's hook. The value of the attribute ARG*N* is the hook variable of the argument. A slightly more complicated example is shown in Figure 3, it features the introduction of an additional proposition and a scope constraint. This rule binds a declarative (marked by the complementizer *dass*) finite clausal object (`oc_fin`) to the verb it is an argument of. To achieve this binding, a proposition relation is assigned as the value of the verb's ARG2, and this proposition in turn has an ARG0,

[2] In fact, for modifiers and specifiers we define lexical RMRSs in a special way, in that we immediately bind the semantic argument. The motivation for this is that whenever one of the dependency relations `mo` or `spec` are encountered, no matter what their exact `Pred` value may be, the semantics contributed by the head of this dependency can be unambiguously related to the semantic head, and is thus recorded already at the "lexical" level.

[3] As explained below, the information about subcategorized arguments is reconstructed from the triples, in the predicate `arg(Fctor,N,Arg)`, where `N` encodes the argument position, and `Fctor` and `Arg` are indices of functor and argument, respectively.

(a)
```
      +arg(X,2,Arg), +g_f(Arg,'oc_fin'),
      get_lb(X,LbX), get_lb(Arg,LbArg),
      +comp_form(Arg,dass),
==>   sort(Lb,h), sort(LbPrpstn,h)
&&    add_ep(LbX,ep_arg2,argx,LbPrpstn)
&&    add_ep(LbPrpstn,ep_rel,rel,'prpstn_m_rel')
&&    add_ep(LbPrpstn,ep_arg0,arg0,Lb)
&&    add_qeq(Lb,LbArg).
```

(b)

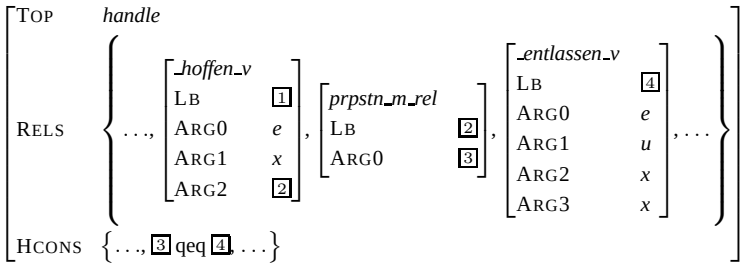

Figure 3: (a) Sample argument binding rule and (b) output RMRS.

which takes scope over the hook label of the matrix verb in the object clause (for the definition of the template `add_ep`, see Figure 2; the template `add_qeq` works similarly: It adds a qeq constraint to the set of handle constraints). In general, the binding of arguments does not depend on the order of rule applications. That is, the fact that the system performs concurrent rule applications in a cascaded rule set is not problematic for semantics construction. Though, we have to make sure that every partial structure is assigned a hook, prior to the application of composition rules. This is ensured by stating the rules for lexical RMRSs first.

Scope constraints. In having the rules introduce handle constraints, we define restrictions on the possible scoped readings. These are defined maximally restrictive in the sense that they must allow for all and only the admissible scopes. This is achieved by gradually adding qeq relations to the global HCONS set. Typically, this constraint relates a handle argument of a scopal element, e. g. a quantifier, and the label of the outscoped element. However, we cannot always fully predict the interaction among several scoping elements. This is the case, inter alia, for the modification of verbs by more than one scopal adverb. This type of ambiguity is modeled by means of a UDRT-style underspecification, that is, we leave the scope among the modifiers unspecified, but restrict each to outscope the verb handle.[4]

3.3 Challenges

Some aspects of semantic composition crucially depend on lexical and phrase structural information which is not available from the dependencies. Here we

[4]This is in accordance with the German HPSG grammar, and will also be adapted in the ERG (p.c. D. Flickinger).

briefly point out the problems and how we solved them.

Argument Structure. While LFG grammars explicitly encode argument structure in the semantic form of the predicate, the derived dependency triples only record the atomic PRED value. We recover the missing information by way of preprocessing rules. The rules make reference to the local grammatical functions of a predicate, and test for features typically borne by non-arguments, for instance, expletives can be identified via the feature `pron_type(_,expl)`. In the composition step, the resulting `arg` predicates will be interpreted as the *slots* that a functor needs to fill.

The TIGER-DB does not provide information about control properties of equiverbs, nor do they mark scopal modifiers. We extracted lexical entries from our broad-coverage German HPSG grammar, and interleave them with the rules for semantics construction, to ensure their proper representation.

Constituency. It is often assumed that there is a crucial difference between the semantics of VP-modification and that of S-modification. Thus, we are faced with the problem that no distinction whatsoever is drawn between heads and their projections in the dependency structures. Hence, we restrict scope with respect to the verb, but do not exclude the proposition-modifying reading.

Similarly, coordination is represented as a set of conjuncts in the triples, but to meet the binary branching coordination analysis of HPSG, we must construct a recursive semantic embedding of partial coordinations. The rules process the conjuncts in a right-to-left manner, each time combining the partial coordination to the right with the conjunct on the left, thereby building a left-branching coordination.

3.4 Treebank Construction and Quality Control

TIGER 700 RMRS Treebank. Our aim is to construct a treebank of 700 sentences from the TIGER dependency bank. Instead of selecting a random sample of sentences, we opt for a block of consecutive sentences. In this way, the treebank can be further extended by annotations for intersentential phenomena, such as co-reference relations, or discourse relations.

However, we have to accommodate for gaps, due to sentences for which there are reasonable syntactic, but (currently) no sound semantic analyses. This problem arises for sentences involving, e.g., elliptical constructions, or else ungrammatical or fragmented sentences. We include, but explicitly mark such sentences for which we can obtain partial, but no fully sound semantic analyses. We correspondingly extend the annotation set to yield a total of 700 correctly annotated sentences.

Automatic Conversion and Quality Control. For compilation of a manualy controlled RMRS bank, we implemented a cascaded approach for quality control, with an initial feedback loop between (i) and (ii):

(i) Manual phenomenon-based error-detection. In the construction process, we mark the application of construction rules by inserting phenomenon-specific identifiers, and use these to select sample RMRSs for phenomenon-based inspection, both in the development phase and for final quality control.

(ii) Investigation of detected errors can result in the improvement of automatic RMRS construction (feedback loop to (i)). Errors that cannot be covered by general rules need to be adjusted manually.

(iii) Manual control. Finally, we perform manual control and correction of errors that could not be covered by automatic RMRS construction. In this phase, we mark and separate the structures or phenomena that are not covered by the state-of-the-art in RMRS-based semantic theory.

Results. The transfer grammar comprises 74 rewrite rules for converting dependency structures to RMRS, plus 34 macros and templates.

In a first validation experiment on the basis of 100 structures, we classified 20% of the RMRSs as involving errors that can be captured by adjustments of the automatic conversion rules (see step (ii) above), while 59% were fully correct.

After improvement of the rules we evaluated the quality of the automatic construction procedure by validating the 700 sentences of the treebank. Of the 700 structures, 4% contained phenomena which we do not analyse at all. 40% required no correction at all. For the 59% that needed manual correction, the average count of units to be corrected per sentence was 3.75. The number of RMRSs that needed less than the average of corrections was 601, i.e. 85.86%.

4 Conclusion

We have presented a method for semantics construction which converts dependency structures to (R)MRSs as they are output by HPSG grammars. This approach allows cross-framework parser evaluation on a broad-coverage basis, and can be applied to existing dependency banks for English (e. g. King et al. (2003)). As shown by manual correction of the automatically constructed RMRS treebank, our semantics construction method yields high-quality results, and can be extended to a full parsing achitecture. A more extensive description and evaluation of the present work can be found in Spreyer and Frank (2005).

References

Brants, S., Dipper, S., Hansen, S., Lezius, W. and Smith, G. 2002. The TIGER Treebank. In *Proceedings of the Workshop on Treebanks and Linguistic Theories*, Sozopol, Bulgaria.

Carroll, J., Frank, A., Lin, D., Prescher, D. and Uszkoreit, H. (eds.). 2002. *Beyond PARSEVAL – Towards Improved Evaluation Measures for Parsing Systems*,

Workshop Proceedings of the *Third International Conference on Language Resources and Evaluation*, LREC 2002 Conference, Las Palmas, Gran Canaria.

Copestake, A., Lascarides, A. and Flickinger, D. 2001. An Algebra for Semantic Construction in Constraint-based Grammars. In *Proceedings of the ACL 2001*, Toulouse, France.

Copestake, A. 2003. Report on the Design of RMRS. Technical Report D1.1a, University of Cambridge, University of Cambridge, UK., 23 pages.

Copestake, A., Flickinger, D., Sag, I. and Pollard, C. 2005. Minimal Recursion Semantics, to appear.

Crouch, R. 2005. Packed Rewriting for Mapping Semantics to KR. In *Proceedings of the Sixth International Workshop on Computational Semantics, IWCS-06*, Tilburg, The Netherlands.

Crouch, R., Kaplan, R., King, T.H. and Riezler, S. 2002. A comparison of evaluation metrics for a broad coverage parser. In *Beyond PARSEVAL. Workshop at the LREC 2002 Conference*, Las Palmas.

Dalrymple, M. (ed.). 1999. *Semantics and Syntax in Lexical Functional Grammar: The Resource Logic Approach*. MIT Press.

Dyvik, H., Rosén, V. and Meurer, P. 2005. LFG, Minimal Recursion Semantics and Translation. In *Proceedings of the LFG 2005 Conference*, CSLI Publications, to appear.

Forst, M. 2003. Treebank Conversion – Establishing a testsuite for a broad-coverage LFG from the TIGER treebank. In *Proceedings of LINC'03*, Budapest, Hungary.

Forst, M., Bertomeu, N., Crysmann, B., Fouvry, F., Hansen-Schirra, S. and Kordoni, V. 2004. Towards a Dependency-Based Gold Standard for German Parsers: The Tiger Dependency Bank. In S. Hansen-Schirra, S. Oepen and H. Uszkoreit (eds.), *Proceedings of LINC 2004*, Geneva, Switzerland.

Kaplan, R.M., Riezler, S., King, T.H., Maxwell, J.T., Vasserman, A. and Crouch, R. 2004. Speed and Accuracy in Shallow and Deep Stochastic Parsing. In *Proceedings of HLT-NAACL'04*, Boston, MA.

King, T.H., Crouch, R., Riezler, S., Dalrymple, M. and Kaplan, R. 2003. The PARC 700 Dependency Bank. In *Proceedings of LINC 2003*, Budapest.

Spreyer, K. and Frank, A. 2005. The TIGER RMRS Bank: RMRS Construction from Dependencies. In F. Bond, S. Oepen and K. Paic (eds.), *Proceedings of LINC 2005*, pages 1–10, Jeju Island, Korea.