

The Grammar Matrix: An Open-Source Starter-Kit for the Rapid Development of Cross-Linguistically Consistent Broad-Coverage Precision Grammars

Emily M. Bender and Dan Flickinger and Stephan Oepen

Center for the Study of Language and Information

Stanford University

{bender | dan | oe}@csli.stanford.edu

Abstract

The grammar matrix is an open-source starter-kit for the development of broad-coverage HPSGs. By using a type hierarchy to represent cross-linguistic generalizations and providing compatibility with other open-source tools for grammar engineering, evaluation, parsing and generation, it facilitates not only quick start-up but also rapid growth towards the wide coverage necessary for robust natural language processing and the precision parses and semantic representations necessary for natural language understanding.

1 Introduction

The past decade has seen the development of wide-coverage implemented grammars representing deep linguistic analysis of several languages in several frameworks, including Head-Driven Phrase Structure Grammar (HPSG), Lexical-Functional Grammar (LFG), and Lexicalized Tree Adjoining Grammar (LTAG). In HPSG, the most extensive grammars are those of English (Flickinger, 2000), German (Müller & Kasper, 2000), and Japanese (Siegel, 2000; Siegel & Bender, 2002). Despite being couched in the same general framework and in some cases being written in the same formalism and consequently being compatible with the same parsing and generation software, these grammars were developed more or less independently of each other. They each represent between 5 and 15 person years of research efforts, and comprise 35–70,000 lines of code. Unfortunately, most of that research is undocumented and the accumulated analyses, best practices for grammar engineering, and tricks of the trade are only available through painstaking inspection of the grammars and/or consultation with their authors. This lack of documentation holds across frameworks, with certain notable exceptions, in-

cluding Alshawi (1992), Müller (1999), and Butt, King, Niño, & Segond (1999).

Grammars which have been under development for many years tend to be very difficult to mine for information, as they contain layers upon layers of interacting analyses and decisions made in light of various intermediate stages of the grammar. As a result, when embarking on the creation of a new grammar for another language, it seems almost easier to start from scratch than to try to model it on an existing grammar. This is unfortunate—being able to leverage the knowledge and infrastructure embedded in existing grammars would greatly accelerate the process of developing new ones. At the same time, these grammars represent an untapped resource for the bottom-up exploration of language universals.

As part of the LinGO consortium’s multi-lingual grammar engineering effort, we are developing a ‘grammar matrix’ or starter-kit, distilling the wisdom of existing grammars and codifying and documenting it in a form that can be used as the basis for new grammars.

In the following sections, we outline the inventory of a first, preliminary version of the grammar matrix, discuss the interaction of basic construction types and semantic composition in unification grammars by means of a detailed example, and consider extensions to the core inventory that we foresee and an evaluation methodology for the matrix proper.

2 Preliminary Development of Matrix

We have produced a preliminary version of the grammar matrix relying heavily on the LinGO project’s English Resource Grammar, and to a lesser extent on the Japanese grammar developed jointly between DFKI Saarbrücken (Germany) and YY Technologies (Mountain View, CA). This early version of the matrix comprises the following com-

ponents:

- Types defining the basic feature geometry and technical devices (e.g., for list manipulation).
- Types associated with Minimal Recursion Semantics (see, e.g., Copestake, Lascarides, & Flickinger, 2001), a meaning representation language which has been shown to be well-suited for semantic composition in typed feature structure grammars. This portion of the grammar matrix includes a hierarchy of relation types, types and constraints for the propagation of semantic information through the phrase structure tree, a representation of illocutionary force, and provisions for grammar rules which make semantic contributions.
- General classes of rules, including derivational and inflectional (lexical) rules, unary and binary phrase structure rules, headed and non-headed rules, and head-initial and head-final rules. These rule classes include implementations of general principles of HPSG, like, for example, the Head Feature and Non-Local Feature Principles.
- Types for basic constructions such as head-complement, head-specifier, head-subject, head-filler, and head-modifier rules, coordination, as well as more specialized classes of constructions, such as relative clauses and noun-noun compounding. Unlike in specific grammars, these types do not impose any ordering on their daughters in the grammar matrix.

Included with the matrix are configuration and parameter files for the LKB grammar engineering environment (Copestake, 2002).

Although small, this preliminary version of the matrix already reflects the main goals of the project: (i) Consistent with other work in HPSG, semantic representations and in particular the syntax-semantics interface are developed in detail; (ii) the types of the matrix are each representations of generalizations across linguistic objects and across languages; and (iii) the richness of the matrix and the incorporation of files which connect it with the LKB allow for extremely quick start-up as the matrix is applied to new languages.

Since February 2002, this preliminary version of the matrix has been in use at two Norwegian universities, one working towards a broad-coverage

reference implementation of Norwegian (NTNU), the other—for the time being—focused on specific aspects of clause structure and lexical description (Oslo University). In the first experiment with the matrix, at NTNU, basic Norwegian sentences were parsing and producing reasonable semantics within two hours of downloading the matrix files. Linguistic coverage should scale up quickly, since the foundation supplied by the matrix is designed not only to provide a quick start, but also to support long-term development of broad-coverage grammars. Both initiatives have confirmed the utility of the matrix starter kit and already have contributed to a series of discussions on cross-lingual HPSG design aspects, specifically in the areas of argument structure representations in the lexicon and basic assumptions about constituent structure (in one view, Norwegian exhibits a VSO topology in the main clause). The user groups have suggested refinements and extensions of the basic inventory, and it is expected that general solutions, as they are identified jointly, will propagate into the existing grammars too.

3 A Detailed Example

As an example of the level of detail involved in the grammar matrix, in this section we consider the analysis of intersective and scopal modification. The matrix is built to give Minimal Recursion Semantics (MRS; Copestake et al., 2001; Copestake, Flickinger, Sag, & Pollard, 1999; Copestake, Flickinger, Malouf, Riehemann, & Sag, 1995) representations. The two English examples in (1) exemplify the difference between intersective and scopal modification:¹

- (1) a. Keanu studied Kung Fu on a spaceship.
- b. Keanu probably studied Kung Fu.

The MRSs for (1a-b) (abstracting away from agreement information) are given in (2) and (3). The MRSs are ordered tuples consisting of a top handle (*h1* in both cases), an instance or event variable (*e* in both cases), a bag of elementary predications (*eps*), and a bag of scope constraints (in these cases, QEQ constraints or ‘equal modulo quantifiers’). In a well-formed MRS, the handles can be

¹These examples also differ in that *probably* is a pre-head modifier while *on a spaceship* is a post-head modifier. This word-order distinction cross-cuts the semantic distinction, and our focus is on the latter, so we won’t consider the word-order aspects of these examples here.

identified in one or more ways respecting the scope constraints such that the dependencies between the eps form a tree. For a detailed description of MRS, see the works cited above. Here, we will focus on the difference between the intersective modifier *on* (a *spaceship*) and the scopal modifier *probably*.

In (2), the ep contributed by *on* (‘on-rel’) shares its handle (*h7*) with the ep contributed by the verb it is modifying (‘study-rel’). As such, the two will always have the same scope; no quantifier can intervene. Further, the second argument of the on-rel (*e*) is the event variable of the study-rel. The first argument, *e'*, is the event variable of the on-rel and the third argument, *z*, is the instance variable of the spaceship-rel.

(2) { h1, e,
 { h1:prpstn-rel(h2), h3:def-np-rel(x, h4, h5),
 h6:named-rel(x, ‘Keanu’), h7:study-rel(e, x, y),
 h8:def-np-rel(y, h9, h10),
 h11:named-rel(y, ‘Kung Fu’), h7:on-rel(e', e, z),
 h12:a-quant-rel(z, h13, h14),
 h15:spaceship-rel(z) },
 { h2 QEQ h7, h4 QEQ h6, h19 QEQ h11,
 h13 QEQ h15 } }

In (3), the ep contributed by the scopal modifier *probably* (‘probably-rel’) has its own handle (*h7*) which is not shared by anything. Furthermore, it takes a handle (*h8*) rather than the event variable of the study-rel as its argument. *h8* is equal modulo quantifiers (QEQ) to the handle of the study-rel (*h9*), and *h7* is equal modulo quantifiers to the argument of the prpstn-rel (*h2*). The prpstn-rel is the ep representing the illocutionary force of the whole expression. This means that quantifiers associated with the NPs *Keanu* and *Kung Fu* can scope inside or outside *probably*.

(3) { h1, e,
 { h1:prpstn-rel(h2), h3:def-np-rel(x, h4, h5),
 h6:named-rel(x, ‘Keanu’),
 h7:probably-rel(h8), h9:study-rel(e, x, y),
 h10:def-np-rel(y, h11, h12),
 h13:named-rel(y, ‘Kung Fu’) },
 { h2 QEQ h7, h4 QEQ h6, h8 QEQ h9,
 h11 QEQ h13 } }

While the details of modifier placement, which parts of speech can modify which kinds of phrases, etc., differ across languages, we believe that all languages display a distinction between scopal and intersective modification. Accordingly, the types

```
isect-mod-phrase := head-mod-phr-simple &
[ HEAD-DTR.SYNSEM.LOCAL
  [ CONT [ TOP #hand,
          INDEX #index ],
    KEYS.MESSAGE 0-dlist ],
  NON-HEAD-DTR.SYNSEM.LOCAL
  [ CAT.HEAD.MOD <[ LOCAL isect-mod ]>,
    CONT.TOP #hand ],
  C-CONT.INDEX #index ].
```

Figure 1: TDL description of *isect-mod-phrase*

```
scopal-mod-phrase := head-mod-phr-simple &
[ NON-HEAD-DTR.SYNSEM.LOCAL
  [ CAT.HEAD.MOD <[ LOCAL scopal-mod ]>,
    CONT.INDEX #index ],
  C-CONT.INDEX #index ].
```

Figure 2: TDL description of *scopal-mod-phrase*

necessary for describing these two kinds of modification are included in the matrix.

The types *isect-mod-phrase* and *scopal-mod-phrase* (shown in Figures 1 and 2) encode the information necessary to build up in a compositional manner the modifier portions of the MRSs in (2) and (3).

These types are embedded in the type hierarchy of the matrix. Through their supertype *head-mod-phr-simple* they inherit information common to many types of phrases, including the basic feature geometry, head feature and non-local feature passing, and semantic compositionality. These types also have subtypes in the matrix specifying the two word-order possibilities (pre- or post-head modifiers), giving a total of four subtypes.²

The most important difference between these types is in the treatment of the handle of the head daughter’s semantics, to distinguish intersective and scopal modification. In *isect-mod-phrase*, the top handles (TOP) of the head and non-head (i.e., modifier) daughters are identified (#hand). This allows for MRSs like (2) where the eps contributed by the head (‘study-rel’) and the modifier (‘on-rel’) take the same scope. The type *scopal-mod-phrase* bears no such constraint. This allows for MRSs like (3) where the modifier’s semantic contribution (‘probably-rel’) takes the handle of the head’s semantics (‘study-rel’) as its argument, so that the modifier outscopes the head. In both types of mod-

²All four subtypes are provided on the theory that most languages will make use of all or most of them.

ifier phrase, a constraint inherited from the super-type ensures that the handle of the modifier is also the handle of the whole phrase.

The constraints on the LOCAL value inside the modifier's MOD value regulate which lexical items can appear in which kind of phrase. Intersective modifiers specify lexically that they are [MOD ⟨ [LOCAL isect-mod] ⟩] and scopal modifiers specify lexically that they are [MOD ⟨ [LOCAL scopal-mod] ⟩].³ These constraints exemplify the kind of information that will be developed in the lexical hierarchy of the matrix.

It is characteristic of broad-coverage grammars that every particular analysis interacts with many other analyses. Modularization is an on-going concern, both for maintainability of individual grammars, and for providing the right level of abstraction in the matrix. For the same reasons, we have only been able to touch on the highlights of the semantic analysis of modification here, but hope that this quick tour will suffice to illustrate the extent of the jump-start the matrix can give in the development of new grammars.

4 Future Extensions

The initial version of the matrix, while sufficient to support some useful grammar work, will require substantial further development on several fronts, including lexical representation, syntactic generalization, sociolinguistic variation, processing issues, and evaluation. This first version drew most heavily from the implementation of the English grammar, with some further insights drawn from the grammar of Japanese. Extensions to the matrix will be based on careful study of existing implemented grammars for other languages, notably German, Spanish and Japanese, as well as feedback from those using the first version of the matrix.

For lexical representation, one of the most urgent needs is to provide a language-independent type hierarchy for the lexicon, at least for major parts of speech, establishing the mechanisms used for linking syntactic subcategorization to semantic predicate-argument structure. Lexical rules provide a second mechanism for expressing general-

izations within the lexicon, and offer ready opportunities for cross-linguistic abstractions for both inflectional and derivational regularities. Work is also progressing on establishing a standard relational database (using PostgreSQL) for storing information for the lexical entries themselves, improving both scalability and clarity compared to the current simple text file representation. Form-based tools will be provided both for constructing lexical entries and for viewing the contents of the lexicon.

The primary focus of work on syntactic generalization in the matrix is to support more freedom in word order, for both complements and modifiers. The first step will be a relatively conservative extension along the lines of Netter (1996), allowing the grammar writer more control over how a head combines with complements of different types, and their interleaving with modifier phrases. Other areas of immediate cross-linguistic interest include the hierarchy of head types, control phenomena, clitics, auxiliary verbs, noun-noun compounds, and more generally, phenomena that involve the word/phrase distinction, such as noun incorporation. A study of the existing grammars for English, German, Japanese, and Spanish reveals a high degree of language-specificity for several of these phenomena, but also suggests promise of reusable abstractions.

Several kinds of sociolinguistic variation require extensions to the matrix, including grammaticized aspects of pragmatics such as politeness and empathy, as well as dialect and register alternations. The grammar of Japanese provides a starting point for representations of both empathy and politeness. Implementations of familiar vs. formal verb forms in German and Spanish provide further instances of politeness to help build the cross-linguistic abstractions. Extensions for dialect variation will build on some exploratory work in adapting the English grammar to support American, British, and Australian regionalisms, both lexical and syntactic, while restricting dialect mixture in generation and associated spurious ambiguity in parsing.

While the development of the matrix will be built largely on the LKB platform, support will also be needed for using the emerging grammars on other processing platforms, and for linking to other packages for pre-processing the linguistic input. Several other platforms exist which can efficiently parse text using the existing grammars, includ-

³Note that there are no further subtypes of LOCAL values beyond *isect-mod* and *scopal-mod*. Since these grammars do not make extensive use of subtypes of LOCAL values, they were available for encoding this distinction. Alternative solutions include positing a new feature.

ing the PET system developed in C++ at Saarland University (Germany) and the DFKI (Callmeier, 2000); the PAGE system developed in Lisp at the DFKI (Uszkoreit et al., 1994); the LiLFeS system developed at Tokyo University (Makino, Yoshida, Torisawa, & Tsujii, 1998), and a parallel processing system developed in Objective C at Delft University (The Netherlands; van Lohuizen, 2002). As part of the matrix package, sample configuration files and documentation will be provided for at least some of these additional platforms.

Existing pre-processing packages can also significantly reduce the effort required to develop a new grammar, particularly for coping with the morphology/syntax interface. For example, the ChaSen package for segmenting Japanese input into words and morphemes (Asahara & Matsumoto, 2000) has been linked to at least the LKB and PET systems. Support for connecting implementations of language-specific pre-processing packages of this kind will be preserved and extended as the matrix develops. Likewise, configuration files are included to support generation, at least within the LKB, provided that the grammar conforms to certain assumptions about semantic representation using the Minimal Recursion Semantics framework.

Finally, a methodology is under development for constructing and using test suites organized around a typology of linguistic phenomena, using the implementation platform of the [incr tsdb()] profiling package (Oepen & Flickinger, 1998; Oepen & Callmeier, 2000). These test suites will enable better communication about current coverage of a given grammar built using the matrix, and serve as the basis for identifying additional phenomena that need to be addressed cross-linguistically within the matrix. Of course, the development of the typology of phenomena is itself a major undertaking for which a systematic cross-linguistic approach will be needed, a discussion of which is outside the scope of this report. But the intent is to seed this classification scheme with a set of relatively coarse-grained phenomenon classes drawn from the existing grammars, then refine the typology as it is applied to these and new grammars built using the matrix.

5 Case Studies

One important part of the matrix package will be a library of phenomenon-based analyses drawn from

the existing grammars and over time from users of the matrix, to provide working examples of how the matrix can be applied and extended. Each case study will be a set of grammar files, simplified for relevance, along with documentation of the analysis, and a test suite of sample sentences which define the range of data covered by the analysis. This library, too, will be organized around the typology of phenomena introduced above, but will also make explicit reference to language families, since both similarities and differences among related languages will be of interest in these case studies. Examples to be included in the first release of this library include numeral classifiers in Japanese, subject pro drop in Spanish, partial-VP fronting in German, and verb diathesis in Norwegian.

6 Evaluation and Evolution

The matrix itself is not a grammar but a collection of generalizations across grammars. As such, it cannot be tested directly on corpora from particular languages, and we must find other means of evaluation. We envision overall evaluation of the matrix based on case studies of its performance in helping grammar engineers quickly start new grammars and in helping them scale those grammars up. Evaluation in detail will be based on automatable deletion/substitution metrics, i.e., tools that determine which types from the matrix get used as is, which get used with modifications, and which get ignored in various matrix-derived grammars. Furthermore, if the matrix evolves to include defeasible constraints, these tools will check which constraints get overridden and whether the value chosen is indeed common enough to be motivated as a default value. This evaluation in detail should be paired with feedback from the grammar engineers to determine why changes were made.

The main goal of evaluation is, of course, to improve the matrix over time. This raises the question of how to propagate changes in the matrix to grammars based on earlier versions. The following three strategies (meant to be used in combination) seem promising: (i) segregate changes that are important to sync to (e.g., changes that affect MRS outputs, fundamental changes to important analyses), (ii) develop a methodology for communicating changes in the matrix, their motivation and their implementation to the user community, and (iii) develop tools for semi-automating resynching

of existing grammars to upgrades of the matrix. These tools could use the type hierarchy to predict where conflicts are likely to arise and bring these to the engineer's attention, possibly inspired by the approach under development at CSLI for the dynamic maintenance of the LinGO Redwoods treebank (Oepen et al., 2002).

Finally, while initial development of the matrix has been and will continue to be highly centralized, we hope to provide support for proposed matrix improvements from the user community. User feedback will already come in the form of case studies for the library as discussed in Section 5 above, but also potentially in proposals for modification of the matrix drawing on experiences in grammar development. In order to provide users with some cross-linguistic context in which to develop and evaluate such proposals themselves, we intend to provide some sample matrix-derived grammars and corresponding test suites with the matrix. A user could thus make a proposed change to the matrix, run the test suites for several languages using the supplied grammars which draw from that changed matrix, and use `[incr tsdb()]` to determine which phenomena have been affected by the change. It is clear that full automation of this evaluation process will be difficult, but at least some classes of changes to the matrix will permit this kind of quick cross-linguistic feedback to users with only a modest amount of additional infrastructure.

7 Conclusion

This project carries linguistic, computational, and practical interest. The linguistic interest lies in the HPSG community's general bottom-up approach to language universals, which involves aiming for good coverage of a variety of languages first, and leaving the task of what they have in common for later. (Of course, theory building is never purely data-driven, and there are substantive hypotheses within HPSG about language universals.) Now that we have implementations with fairly extensive coverage for a somewhat typologically diverse set of languages, it is a good time to take the next step in this program, working to extract and generalize what is similar across these existing wide-coverage grammars. Moreover, the central role of types in the representation of linguistic generalizations enables the kind of underspecification which is useful for expressing what is common among related lan-

guages while allowing for the further specialization which necessarily distinguishes one language from another.

The computational interest is threefold. First there is the question of what formal devices the grammar matrix will require. Should it include defaults? What about domain union (linearization theory)? The selection and deployment of formal devices should be informed by on-going research on processing schemes, and here the crosslinguistic perspective can be particularly helpful. Where there are several equivalent analyses of the same linguistic phenomena (e.g., morphosyntactic ambiguity or optionality), the choice of analysis can have processing implications that aren't necessarily apparent in a single grammar. Second, having a set of wide-coverage HPSGs with fairly standardized fundamentals could prove interesting for research on stochastic processing and disambiguation, especially if the languages differ in gross typological features such as word order. Finally, there are also computational issues involved in how the grammar matrix would evolve over time as it is used in new grammars. The matrix enables the developer of a grammar for a new language to get a quick start on producing a system that parses and generates with non-trivial semantics, while also building the foundation for a wide-coverage grammar of the language. But the matrix itself may well change in parallel with the development of the grammar for a particular language, so appropriate mechanisms must be developed to support the merging of enhancements to both.

There is also practical industrial benefit to this project. Companies that are consumers of these grammars benefit when grammars of multiple languages work with the same parsing and generation algorithms and produce standardized semantic representations derived from a rich, linguistically motivated syntax-semantics interface. More importantly, the grammar matrix will help to remove one of the primary remaining obstacles to commercial deployment of grammars of this type and indeed of the commercial use of deep linguistic analysis: the immense cost of developing the resource.

Acknowledgements

Since the grammar matrix draws on prior research and existing grammars, it necessarily reflects contributions from many people. Rob Malouf, Jeff Smith, John Beavers, and Kathryn

Campbell-Kibler have contributed to the LinGO ERG; Melanie Siegel is the original developer for the Japanese grammar. Tim Baldwin, Ann Copestake, Ivan Sag, Tom Wasow, and other members of the LinGO Laboratory at CSLI have had a great deal of influence on the design of the grammatical analyses and corresponding MRS representations. Warmest thanks to Lars Hellan and his colleagues at NTNU and Jan Tore Lønning and his students at Oslo University for their cooperation, patience, and tolerance.

References

- Alshawi, H. (Ed.). (1992). *The Core Language Engine*. Cambridge, MA: MIT Press.
- Asahara, M., & Matsumoto, Y. (2000). Extended models and tools for high-performance part-of-speech tagger. In *Proceedings of the 18th International Conference on Computational Linguistics* (pp. 21–27). Saarbrücken, Germany.
- Butt, M., King, T. H., Niño, M.-E., & Segond, F. (1999). *A grammar writer's cookbook*. Stanford, CA: CSLI Publications.
- Callmeier, U. (2000). PET — A platform for experimentation with efficient HPSG processing techniques. *Natural Language Engineering, 6 (1) (Special Issue on Efficient Processing with HPSG)*, 99–108.
- Copestake, A. (2002). *Implementing typed feature structure grammars*. Stanford, CA: CSLI Publications.
- Copestake, A., Flickinger, D., Malouf, R., Riehemann, S., & Sag, I. (1995). Translation using minimal recursion semantics. In *Proceedings of the Sixth International Conference on Theoretical and Methodological Issues in Machine Translation*. Leuven, Belgium.
- Copestake, A., Flickinger, D. P., Sag, I. A., & Pollard, C. (1999). *Minimal Recursion Semantics. An introduction*. in preparation, CSLI Stanford, Stanford, CA.
- Copestake, A., Lascarides, A., & Flickinger, D. (2001). An algebra for semantic construction in constraint-based grammars. In *Proceedings of the 39th Meeting of the Association for Computational Linguistics*. Toulouse, France.
- Flickinger, D. (2000). On building a more efficient grammar by exploiting types. *Natural Language Engineering, 6 (1) (Special Issue on Efficient Processing with HPSG)*, 15–28.
- van Lohuizen, M. (2002). Efficient and thread-safe unification with LinGO. In S. Oepen, D. Flickinger, J. Tsujii, & H. Uszkoreit (Eds.), *Collaborative language engineering. A case study in efficient grammar-based processing*. Stanford, CA: CSLI Publications. (forthcoming)
- Makino, T., Yoshida, M., Torisawa, K., & Tsujii, J. (1998). LiLFeS — towards a practical HPSG parser. In *Proceedings of the 17th International Conference on Computational Linguistics and the 36th Annual Meeting of the Association for Computational Linguistics* (pp. 807–11). Montreal, Canada.
- Müller, S. (1999). *Deutsche syntax deklarativ. Head-Driven Phrase Structure Grammar für das Deutsche*. Tübingen, Germany: Max Niemeyer Verlag.
- Müller, S., & Kasper, W. (2000). HPSG analysis of German. In W. Wahlster (Ed.), *Verbmobil. Foundations of speech-to-speech translation* (Artificial Intelligence ed., pp. 238–253). Berlin, Germany: Springer.
- Netter, K. (1996). *Functional categories in an HPSG for German*. Unpublished doctoral dissertation, Saarland University, Saarbrücken, Germany.
- Oepen, S., & Callmeier, U. (2000). Measure for measure: Parser cross-fertilization. Towards increased component comparability and exchange. In *Proceedings of the 6th International Workshop on Parsing Technologies* (pp. 183–194). Trento, Italy.
- Oepen, S., & Flickinger, D. P. (1998). Towards systematic grammar profiling. Test suite technology ten years after. *Journal of Computer Speech and Language, 12 (4) (Special Issue on Evaluation)*, 411–436.
- Oepen, S., Toutanova, K., Shieber, S., Manning, C., Flickinger, D., & Brants, T. (2002). The LinGO Redwoods treebank. Motivation and preliminary applications. In *Proceedings of the 19th International Conference on Computational Linguistics*. Taipei, Taiwan.
- Siegel, M. (2000). HPSG analysis of Japanese. In W. Wahlster (Ed.), *Verbmobil. Foundations of speech-to-speech translation* (Artificial Intelligence ed., pp. 265–280). Berlin, Germany: Springer.
- Siegel, M., & Bender, E. M. (2002). Efficient deep processing of Japanese. In *Proceedings of the 19th International Conference on Computational Linguistics*. Taipei, Taiwan.
- Uszkoreit, H., Backofen, R., Busemann, S., Diagne, A. K., Hinkelman, E. A., Kasper, W., Kiefer, B., Krieger, H.-U., Netter, K., Neumann, G., Oepen, S., & Spackman, S. P. (1994). DISCO — an HPSG-based NLP system and its application for appointment scheduling. In *Proceedings of the 15th International Conference on Computational Linguistics*. Kyoto, Japan.