The LinGO Grammar Matrix Customization System

Guest Lecture: Computational Resources and Low Resource Languages

Antske Fokkens

Department of Computational Linguistics
Saarland University

16 November 2011

Outline

Acknowledgments

- This lecture represents joint work with:

  Emily M. Bender, Scott Drellishak, Michael Goodman, Laurie Poulson and Safiyyah Saleem
- This material is based upon work supported by the National Science Foundation under Grant No. 0644097. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

Outline

The Matrix Customization System

The LinGO Matrix Customization System is a tool that provides start-up implementations for linguistically motivated precision grammars

- From an engineering point of view, it supports code-sharing leading to
  - significant reduction of effort in grammar engineering
  - more consistency across grammars
- From a scientific point of view
  - it supports syntactic research
  - it encourages research that combines typological research with formal syntactic analyses

“Deep” or “Precision” Grammars

- **Deep** grammars: parsing leads to, and generation comes from a semantic representation
- **Precision** grammars: they are linguistically based and aim at getting (only) right analyses
- They are constraint based, resulting in relatively low ambiguity, and less robustness
- Linguistic encoding requires manual effort by an expert: they are expensive to build

Multilingual Grammar Engineering

Main Ideas:

- Reduce the efforts of creating new grammars by using knowledge from those already created
- Create consistency between grammars of different languages
- Research on crosslinguistic similarity

These aims also form the main motivation for the Grammar Matrix Customization Project

Why grammar engineering?

- Broad-coverage precision grammars can be used for implementations
  - These grammars provide more elaborate analyses and are more domain independent than statistically trained parsers
  - Little requirements to get started: one grammar engineer and a good descriptive grammar and/or access to native speakers
- Hypothesis testing in syntactic research
- Multi-lingual grammar engineering can support typological research

Outline

Grammar Matrix Context: DELPH-IN

- Delph-in (www.delph-in.net) is a collaboration effort for researchers working on deep linguistic processing.
- The Delph-in member sites contribute open-source software and linguistic resources
- The reference formalism used in Delph-in is based on HPSG (Pollard and Sag (1994)) and use MRS (Copestake et al. (2005)) as parse output and basis for generation
- (Most) grammars are written in tdl (type description language) — interpreted by lkb and PET
- [incr tsdb()] Oepen (2001) for regression testing and treebanking
- Large and medium scale grammars: ERG (English), JACY (Japanese), GG (German), Spanish, NorSource (Norwegian), BURGER (Bulgarian), Russian, Modern Greek, Mandarin Chinese, French

Outline

Grammar Matrix History

- 2001: First-pass cross-linguistic core grammar (Bender et al. (2002))
  - Context: EU Project DeepThought, which included multilingual grammar development
  - Source: English Resource Grammar (Flickinger (2000)), with reference to JACY Japanese Grammar (Siegel and Bender (2002))
  - “Bottom up approach to linguistic universals”: Incremental refinement of core grammar as it gets deployed in different languages

The Grammar Matrix

- The core grammar is encoded in a set of files that can be shared by all Matrix grammars
- The files provide basic implementations of types that are inherited by the individual grammars
- Its contributions are: Feature geometry, semantic compositionality, headedness, head-argument and head-modifier constructions; collateral files for software interaction
- 2002-: Used in development of Norwegian (Hellan and Haugereid (2003)), Modern Greek (Kordoni and Neu (2005)), Spanish (Marimon et al. (2007)) and Italian grammars

The Matrix Core

- The Core Grammar *matrix.tdl* is meant to be used as the basis of all Matrix Grammars. It provides:
  1. Basic features and devices used in HPSG grammars (e.g. phrase, word, category, lists)
  2. Basic grammar rules (e.g. unary/binary rules, head-subject/head-complement/head-specifier, head-final/head-initial)
  3. Basics for semantics: respects principle of semantic compositionality, supports Minimal Recursive Semantics (Copestake et al. (2005))
  4. Some more advanced features (e.g. simple part of speech inventory, argument extraction, coordination)
  5. Language specific grammars can inherit implementations from *matrix.tdl*

The Matrix Core, Example

Implementation for a language with word order **S**ubject **O**bject **V**erb:

comp-head-rule := basic-head-compl-phrase & head-final.

subj-head-rule := basic-head-subj-rule & head-final &
[SYNSEM.LOCAL.VAL.COMPS $< >$].

The basic properties of these rules are defined in *matrix.tdl*.

For comparison: the basic-head-comp-phrase

basic-head-comp-phrase := head-nexus-phrase & basic-binary-headed-phrase &
[SYNSEM *phr-synsem-min &*
[LOCAL [CAT [VAL [SUBJ #*subj*,
SPR #*spr*],
POSTHEAD #*ph*,
HC-LIGHT #*light*],
CONT.HOOK #*hook*],
LIGHT #*light*,
NON-LOCAL.SLASH #*slash*]
INFLECTED +,
HEAD-DTR.SYNSEM [local.cat [VAL [SUBJ #*subj*,
SPR #*spr*],
HC-LIGHT #*light*,
POSTHEAD #*ph*]],
NON-LOCAL.SLASH #*slash*
NON-HEAD-DTR.SYNSEM *canonical-synsem &*
[LOCAL.COORD -],
C-CONT [RELS <! !>,
HCONS <! !>,
HOOK #*hook*],
ARGS < [INFLECTED +],
[INFLECTED +] >].

Grammar Matrix: History

- 2004: First annual multilingual grammar engineering course (Bender 2007)
- Each student works with a different language
- Extend core grammar to different languages, covering:
- Case, agreement, modification, sentential negation, yes-no questions, sentential complements, modals
- Lab instructions outline analyses for known variations

Grammar Matrix: History

- 2005: First pass customization system (Bender and Flickinger 2005)

  Lab instructions were becoming specific enough that a machine could follow them: for some parts only a typological description of a language was necessary

- 2005-2011: Refinements to customization system (Drellishak and Bender (2005), Drellishak (2009), Bender et al. (2010), Saleem and Bender (2010))

Matrix Libraries

- The Matrix Libraries provide implementations of grammar fragments of phenomena that vary cross-linguistically (e.g. word order, case)
- A web-based questionnaire elicits typological descriptions, which evoke specific implementations from the Matrix Libraries
- With the libraries, the customization system output grammar fragments: these can be evaluated!

Outline

Overview of the Matrix System

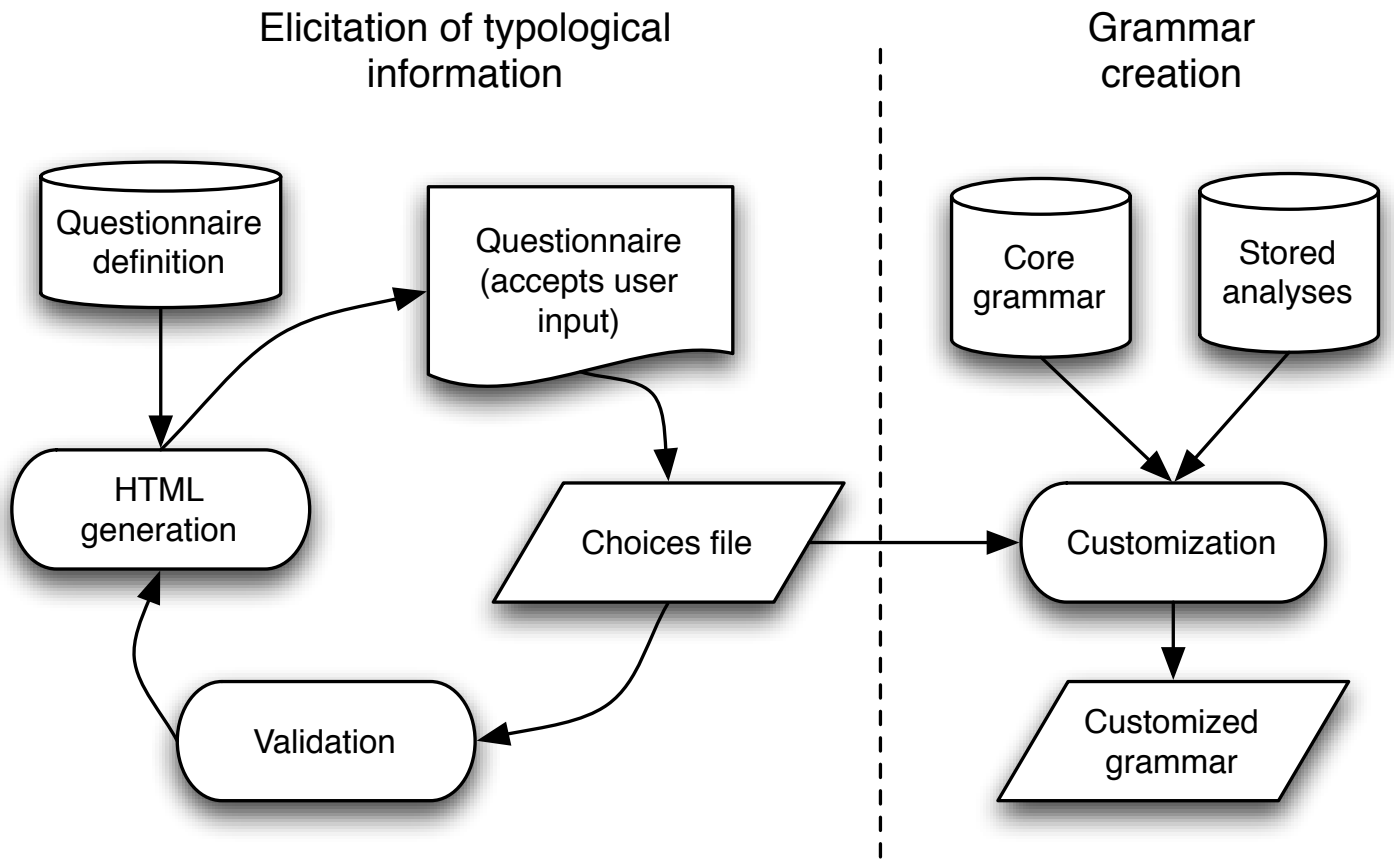


Figure: Schematic system overview

HPSG: A system of signs

- Saussurean signs pair forms (orthographic/phonologica as well as morphosyntactic) with meanings (semantics)
- Lexical entries, lexical rules and phrase struture rules are all signs.
- Signs are modeled with typed feature structures, where features can take on atomic as well as complex values (other feature structures, lists of feature structures)

Unification and typed feature structures

- Feature structures are combined using *unification*.
- Unification is order-independent.
- Types are arranged into a multiple inheritance hierarchy.
- Type constraints specify appropriate features for each type as well as constraints on their values.
- The type hierarchy determines which types will unify with each other (closed-world assumption).

Customized Starter Grammars

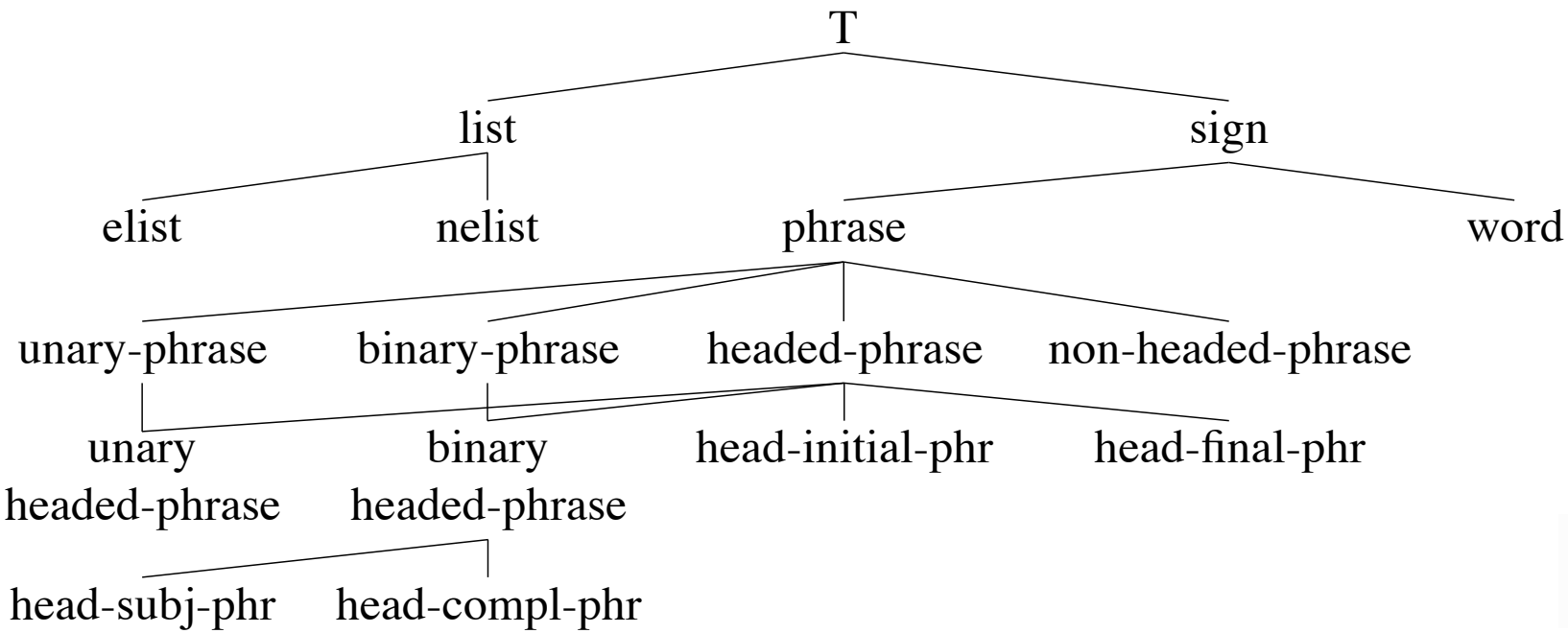
- The customization system provides initial implementations for new grammars
- These ‘starter grammars’ contain the top of the type hierarchy for the language in question
- The initial hierarchy consists of a multilingual core component and basic language specific implementations

The Matrix Core

- The Core Grammar *matrix.tdl* is meant to be used as the basis of all Matrix Grammars. It provides:
  1. Basic features and devices used in HPSG grammars (e.g. phrase, word, category, lists)
  2. Basic grammar rules (e.g. unary/binary rules, head-subject/head-complement/head-specifier, head-final/head-initial)
  3. Basics for semantics: respects principle of semantic compositionality, supports Minimal Recursion Semantics
  4. Some more advanced features (e.g. simple part of speech inventory, argument extraction, coordination)
  5. Language specific grammars can inherit implementations from *matrix.tdl*

The Matrix Core, part of the hierarchy

The Matrix Core provides basic types for cross-linguistic HPSG analyses:



Matrix Libraries

- The Matrix Libraries provide implementations of grammar fragments of phenomena that vary cross-linguistically (e.g. word order, case)
- A web-based questionnaire elicits typological descriptions, which evoke specific implementations from the Matrix Libraries
- Some libraries offer closed menus of preset choices, others offer more flexibility (“metamodeling”).

Starter Grammar

The Matrix Core:

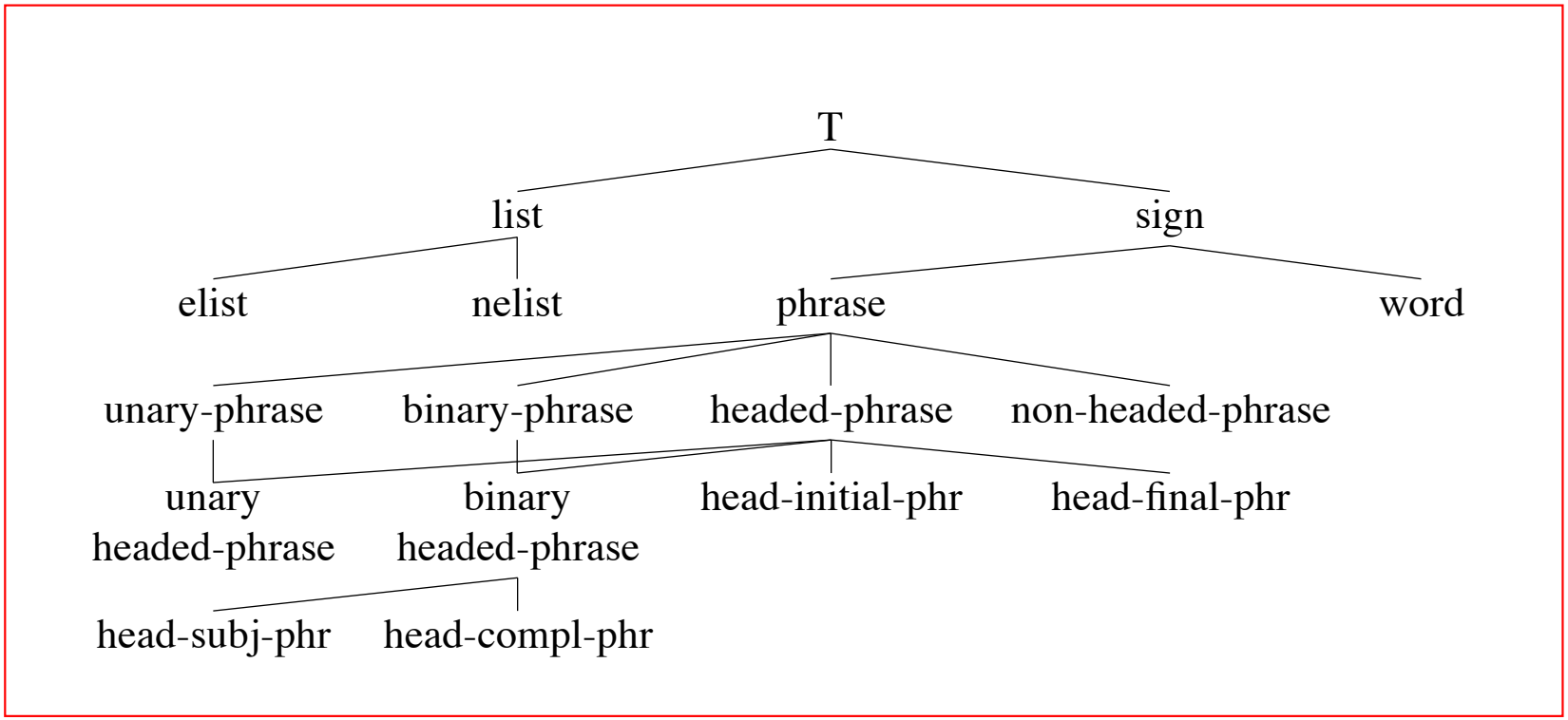
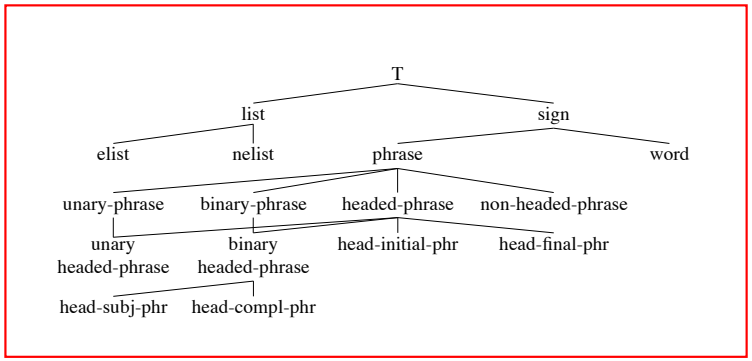


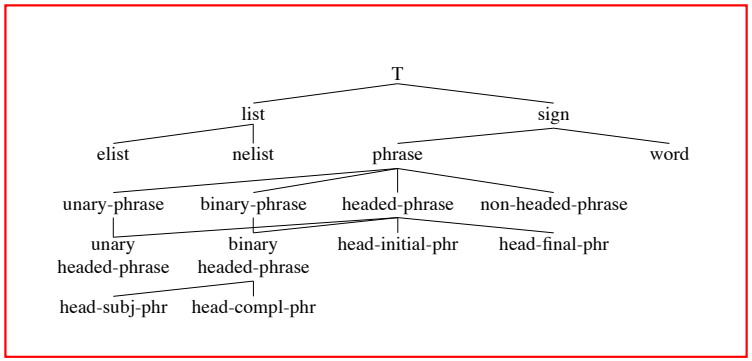
Figure: A small part of the Matrix type hierarchy

Starter Grammar



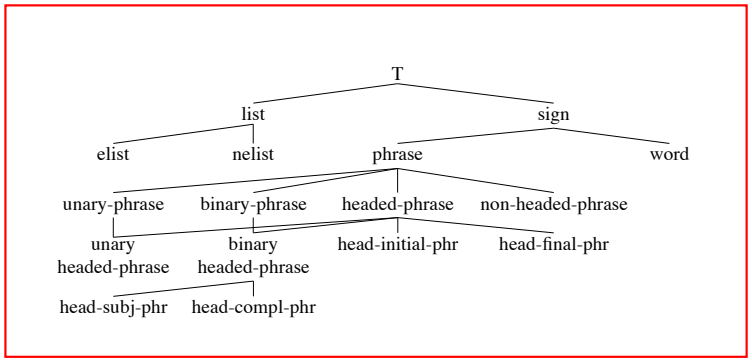
CHOICES

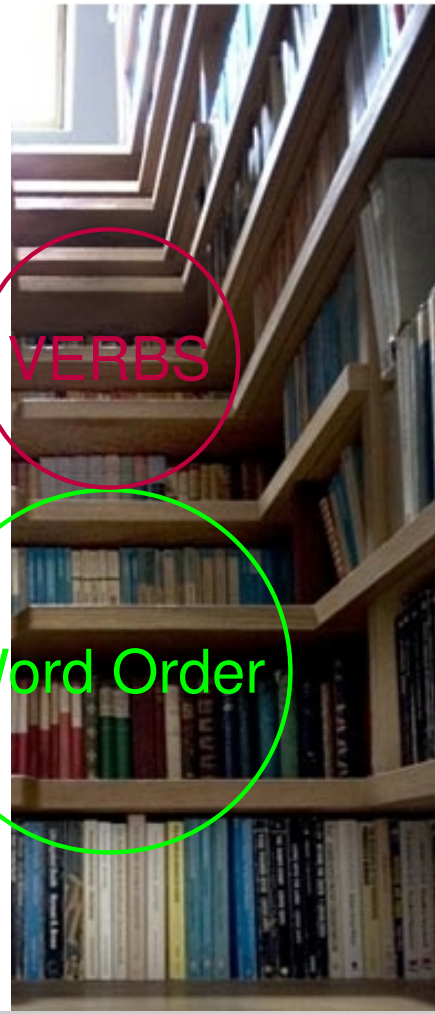
Starter Grammar



CHOICES

Starter Grammar





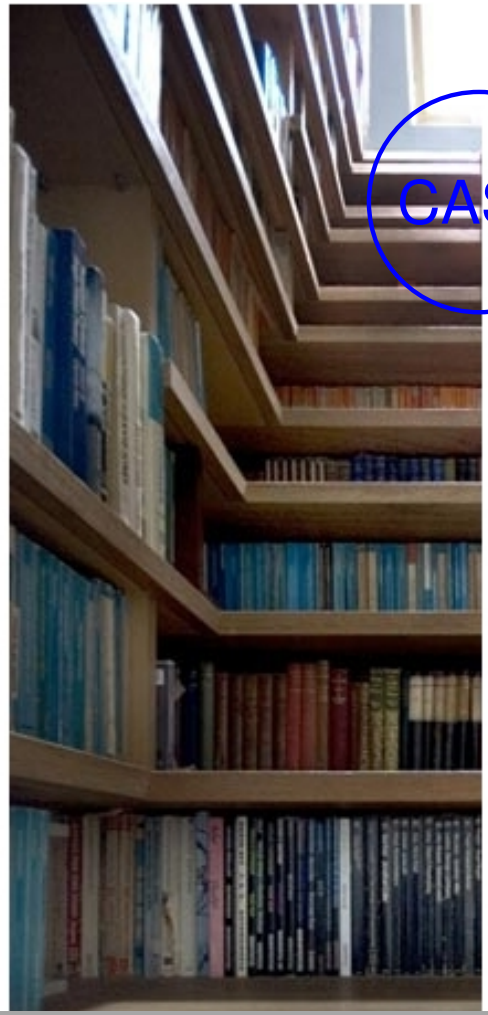
VERBS

CASE

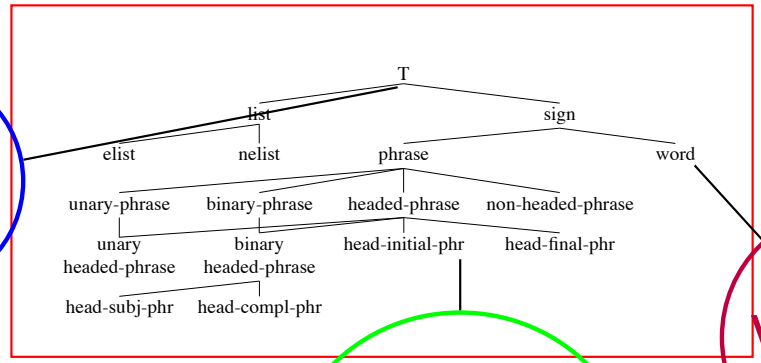
Word Order

CHOICES

Starter Grammar



CASE

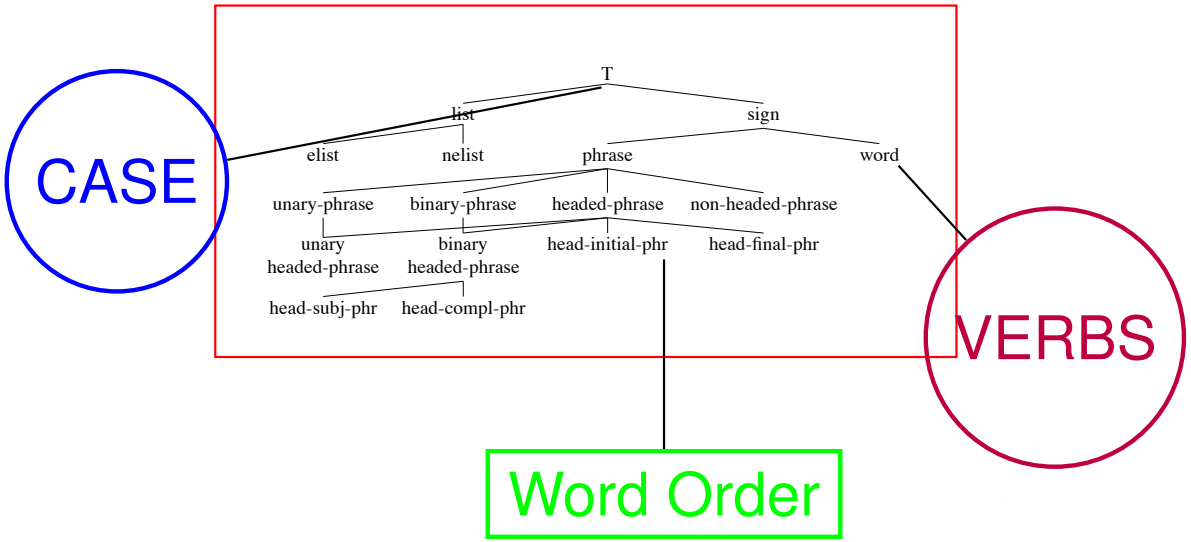


VERBS

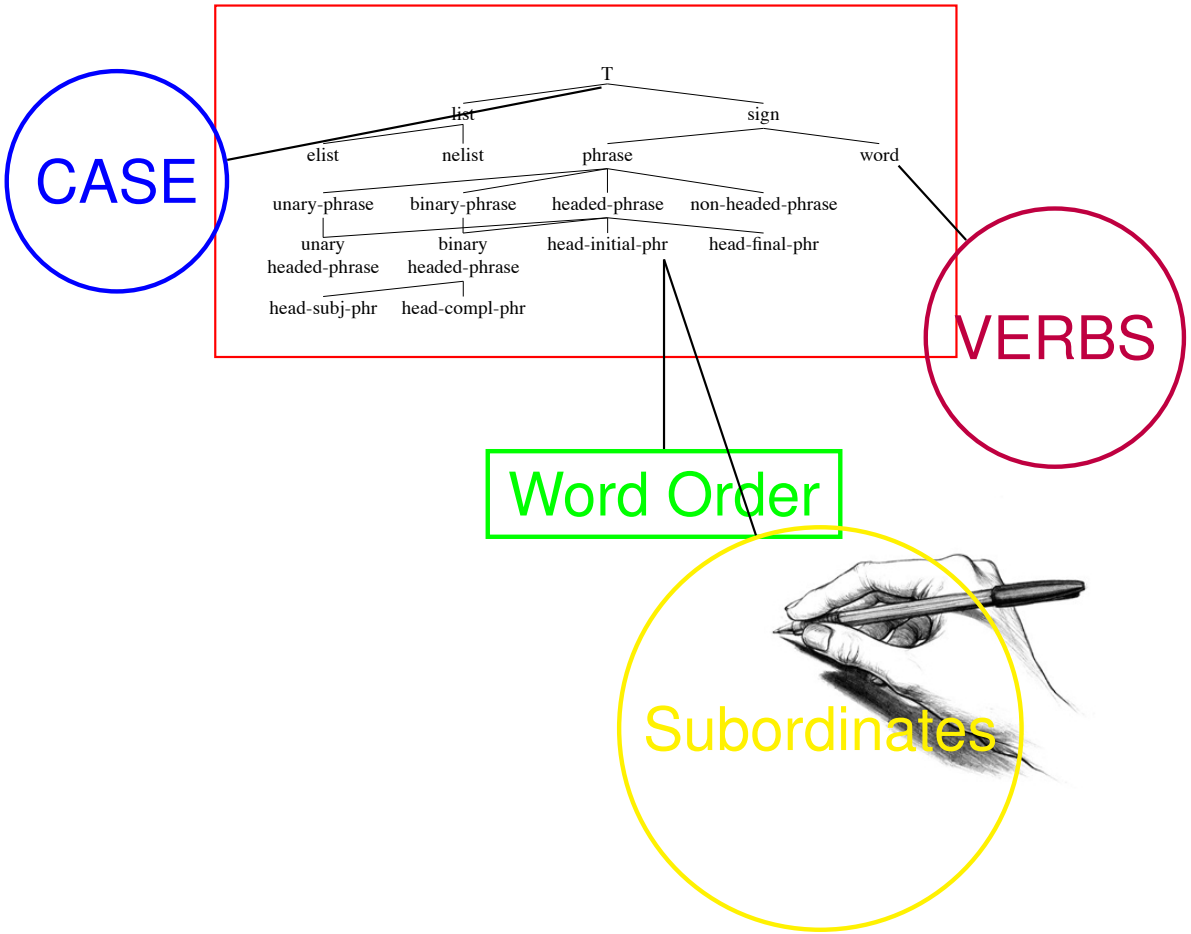
Word Order

CHOICES

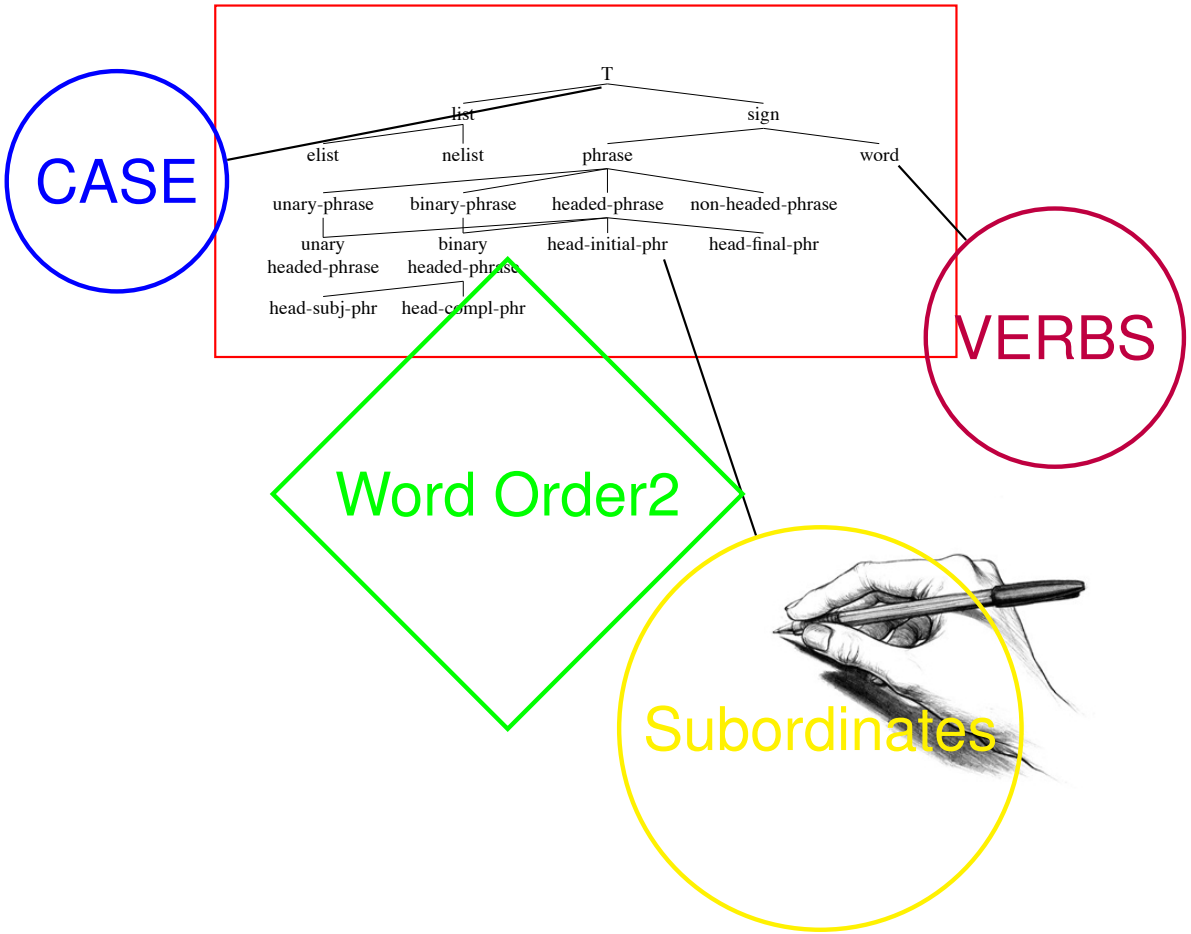
Grammar development



Grammar development



Grammar development



Outline

Libraries and their Foundations

- Libraries based on typological research: person, number, gender, case, coordination (Drellishak 2010), argument optionality (Saleem 2010), Negation (Crowgey, under development)
- Libraries that permit modeling by users: Tense Aspect Mood, Other Features (Poulson), Lexicon and Morphology (O'Hara 2008, Goodman)
- Libraries using basic linguistic knowledge: Direct-Inverse (Drellishak 2010), Word Order (Fokkens 2010), Yes/No Questions (Bender)

Typological variation: width and depth

- The Grammar Matrix customization system aims at covering a wide typological variation
- Aim: a typologocally motivated library should cover basic analyses for all known systems
- Challenge: Typological studies tend to be about global properties:
  $\Rightarrow$ How to decide on the details?
- New tendency: Matrix-related projects that look at detailed implementation and variation for a small set of languages:
  - SlaviGram (Avgustinova and Zhang (2009))
  - CLIMB Germanic grammars (Fokkens (2011))

Sharing more information across languages?

- Getting information from individual languages back to Matrix developers:
  - Intensive collaboration between grammar developer with Matrix developer(s), e.g. Borisova's MSc thesis on Georgian polypersonal agreement
  - CLIMB: Libraries of implementations for individual languages

Matrix Speed-up

- How much does the Grammar Matrix help?
- One could measure speed versus coverage
- But, too many influencing factors:
  - the language itself
  - knowledge about the language (descriptive grammar, syntactic analyses)
  - the grammar engineer
  - the experience of the grammar engineer
- Current question: can a core grammar (to be trained on a Treebank) be created within a year?

Summary

- The Grammar Matrix supports grammar engineers starting a new grammar
- It provides basic implementations for a wide range of typological variation
- Small grammars can now be created with little effort
- Feedback from individual language and evaluation of the practical contribution is part of current research

Bibliography I

Avgustinova, T. and Zhang, Y. (2009). Parallel grammar engineering for Slavic languages. In *Workshop on Grammar Engineering Across Frameworks*, Singapore, Singapore.

Bender, E. M., Drellishak, S., Fokkens, A., Poulson, L., and Saleem, S. (2010). Grammar customization. *Research on Language & Computation*, 8(1):23–72.

Bender, E. M., Flickinger, D., and Oepen, S. (2002). The grammar matrix: An open-source starter-kit for the rapid development of cross-linguistically consistent broad-coverage precision grammars. In Carroll, J., Oostdijk, N., and Sutcliffe, R., editors, *Proceedings of the Workshop on Grammar Engineering and Evaluation at the 19th International Conference on Computational Linguistics*, pages 8–14, Taipei, Taiwan.

Copestake, A., Flickinger, D., Pollard, C., and Sag, I. (2005). Minimal recursion semantics. an introduction. *Journal of Research on Language and Computation*, 3(2–3):281 – 332.

Drellishak, S. (2009). *Widespread But Not Universal: Improving the Typological Coverage of the Grammar Matrix*. PhD thesis, University of Washington.

Drellishak, S. and Bender, E. M. (2005). A coordination module for a crosslinguistic grammar resource. In Müller, S., editor, *The Proceedings of the 12th International Conference on Head-Driven Phrase Structure Grammar, Department of Informatics, University of Lisbon*, pages 108–128, Stanford. CSLI Publications.

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.

Fokkens, A. (2011). Metagrammar engineering: Towards systematic exploration of implemented grammars. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 1066–1076, Portland, Oregon, USA. Association for Computational Linguistics.

Bibliography II

Hellan, L. and Haugereid, P. (2003). NorSource: An exercise in Matrix grammar-building design. In Bender, E. M., Flickinger, D., Fouvry, F., and Siegel, M., editors, *Proceedings of the Workshop on Ideas and Strategies for Multilingual Grammar Development, ESSLLI 2003*, pages 41–48, Vienna, Austria.

Kordoni, V. and Neu, J. (2005). Deep analysis of Modern Greek. In Su, K.-Y., Tsujii, J., and Lee, J.-H., editors, *Lecture Notes in Computer Science*, volume 3248, pages 674–683. Springer-Verlag, Berlin.

Marimon, M., Bel, N., and Seghezzi, N. (2007). Test-suite construction for a Spanish grammar. In King, T. H. and Bender, E. M., editors, *Proceedings of the GEAF 2007 Workshop*, Stanford, CA. CSLI Publications.

Oepen, S. (2001). [incr tsdb()] — competence and performance laboratory. Technical report, DFKI, Saarbrücken, Germany.

Pollard, C. and Sag, I. (1994). *Head-Driven Phrase Structure Grammar*. University of Chicago Press, Chicago, USA.

Saleem, S. and Bender, E. M. (2010). Argument optionality in the lingo grammar matrix. In *Coling 2010: Posters*, pages 1068–1076, Beijing, China. Coling 2010 Organizing Committee.

Siegel, M. and Bender, E. M. (2002). Efficient deep processing of Japanese. In *Proceedings of the 3rd Workshop on Asian Language Resources and International Standardization at the 19th International Conference on Computational Linguistics*, Taipei, Taiwan.