Grammar engineering with HPSG and the Grammar Matrix

Guest lecture for LING566

Olga Zamaraeva
olzama@uw.edu

Department of Linguistics, University of Washington

November 26 2019

GE: Desiderata

GE for Ling. hypoth. test.

The Grammar Matrix

LING567

Open quest. in GE

References

The Grammar Engineering discipline

- Desiderata
- Linguistic hypothesis testing
- The Grammar Matrix: Meta-grammar engineering
- The LKB: A grammar engineering environment
- LING567: A grammar engineering class
- Open questions in grammar engineering

Grammar engineering desiderata

- GE: Modeling **formal** rules of human languages in a machine-readable way
- *Grammar*: Parser/generator which encodes language rules according to a linguistic (e.g. syntactic) theory
  - More precisely, a grammar is input to a parsing/generating algorithtm
- Cross-linguistic: The same core (principles of structure) should be applicable to any language
- Precise: Parses have meaningful syntactic and semantic structure; ungrammatical sentences should not be possible

Grammar engineering's role today

- GE used to be a dominant paradigm in NLP, MT; it isn't today
- It is prominent in tasks such as grammar checking
  - Where precision is important
  - ...though the theoretical groundedness may be deemphasized
- For me personally, GE is at the heart of today's computational linguistics*
- *if CompLing is to be construed as answering linguistic questions with computational means

Two linguistic questions

- *'Are question phrases in this language obligatorily fronted?'*
- *'Can question word fronting for any language be modeled using the same filler-gap rule?'*
- Q: What is the difference between the two questions above?

Two linguistic hypotheses

- *'Question phrases in this language are obligatorily fronted'*
- *'Question word fronting for any language can be modeled using the same filler-gap rule'*
- How do we test them?

Linguistic hypotheses in syntax

- A set of 'elegant' principles, rules, and lexical item structures which together generate only grammatical sentences in a language

Sir Galahad, the Quest for the Holy Grail, by Arthur Hughes (1870)

Formal vs. data-driven approaches (an old debate?)

- Should we hypothesize an idea of human language first or should that idea emerge from the data?

Rafael "The School of Athens"

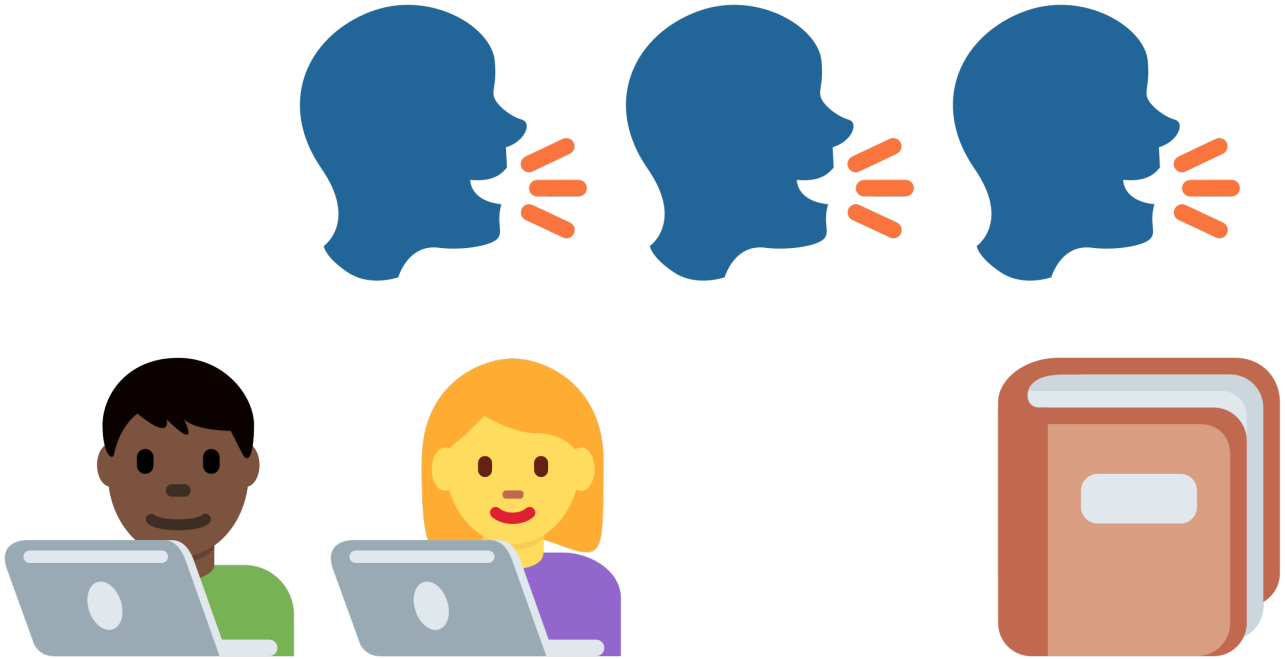
Grammar engineering uses in linguistics

- Linguistics hypothesis testing
  - *'Question phrases in this language are obligatorily fronted'*
  - This is both a **typological** and a **syntactic** hypothesis
  - It is descriptive rather than formal-syntactic
  - How and why would we want a computer to test it?

Grammar engineering uses in linguistics

- Linguistics hypothesis testing
  - *'Question phrases in this language are obligatorily fronted'*
  - This is both a **typological** and a **syntactic** hypothesis
  - It is descriptive rather than formal-syntactic
  - How and why would we want a computer to test it?
- Digital representation of *a set of hypotheses*
  - A grammar such as in the appendix of your book is a documentation of English
  - By loading it into the computer, you can check its consistency, coverage, etc.
  - This is invaluable for endangered language documentation
  - NB: a digital grammar is not a replacement for a prose grammar (a book)
    - why not?

Field Linguistics: Descriptive Grammars



Field Linguistics: Some Challenges

- Accounting for **all** of the data
- ...manually and by introspection

Grammar Engineering: Implemented Grammars

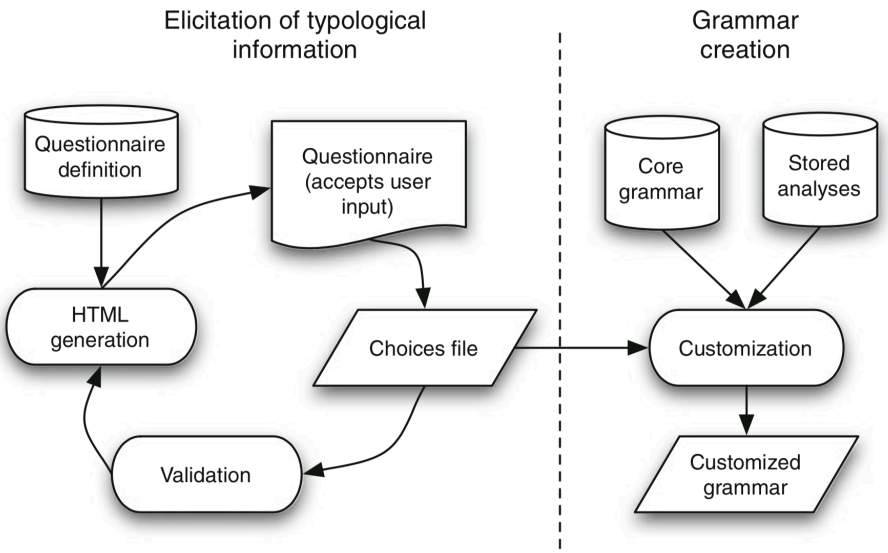
By batch parsing grammatical and ungrammatical sentences, examine the coverage of a grammar which encodes your analyses.

Grammar Engineering: Implemented Grammars

The problem is... Implementing a grammar is COMPLEX.

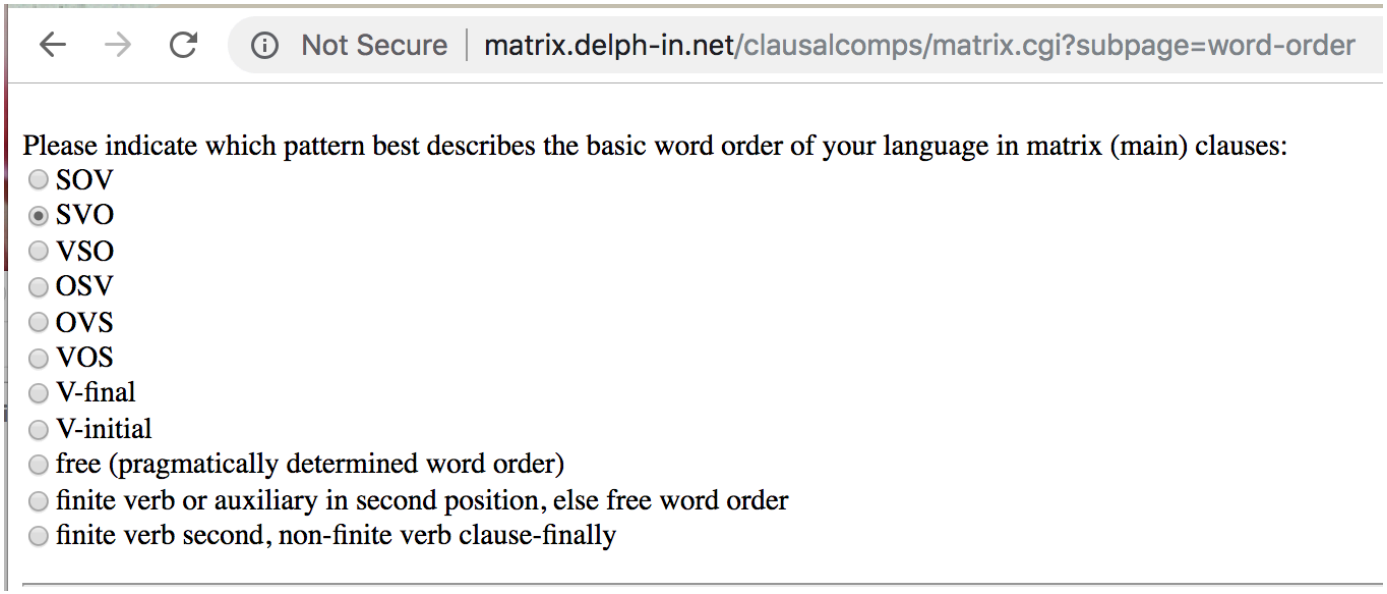
Meta-Grammar Engineering System: The Grammar Matrix (Bender et al. 2010)

- A grammar engineering toolkit
  - A grammar that reflects your hypotheses can be obtained automatically



The Grammar Matrix: Questionnaire (Bender et al. 2002, 2010)

The user fills out typological, lexical, and morphological questionnaire, e.g. the Word Order page:

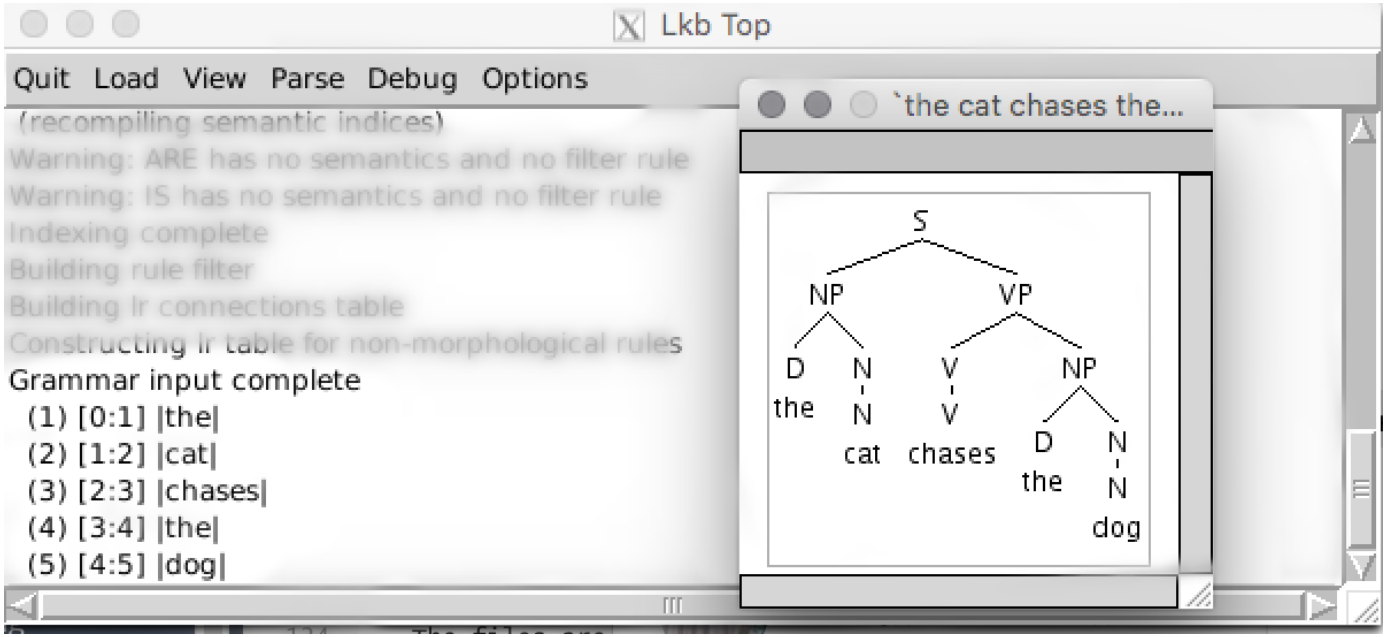


- ...hits a button:

- ...and obtains a folder with some files in it (lexical entries and rules)

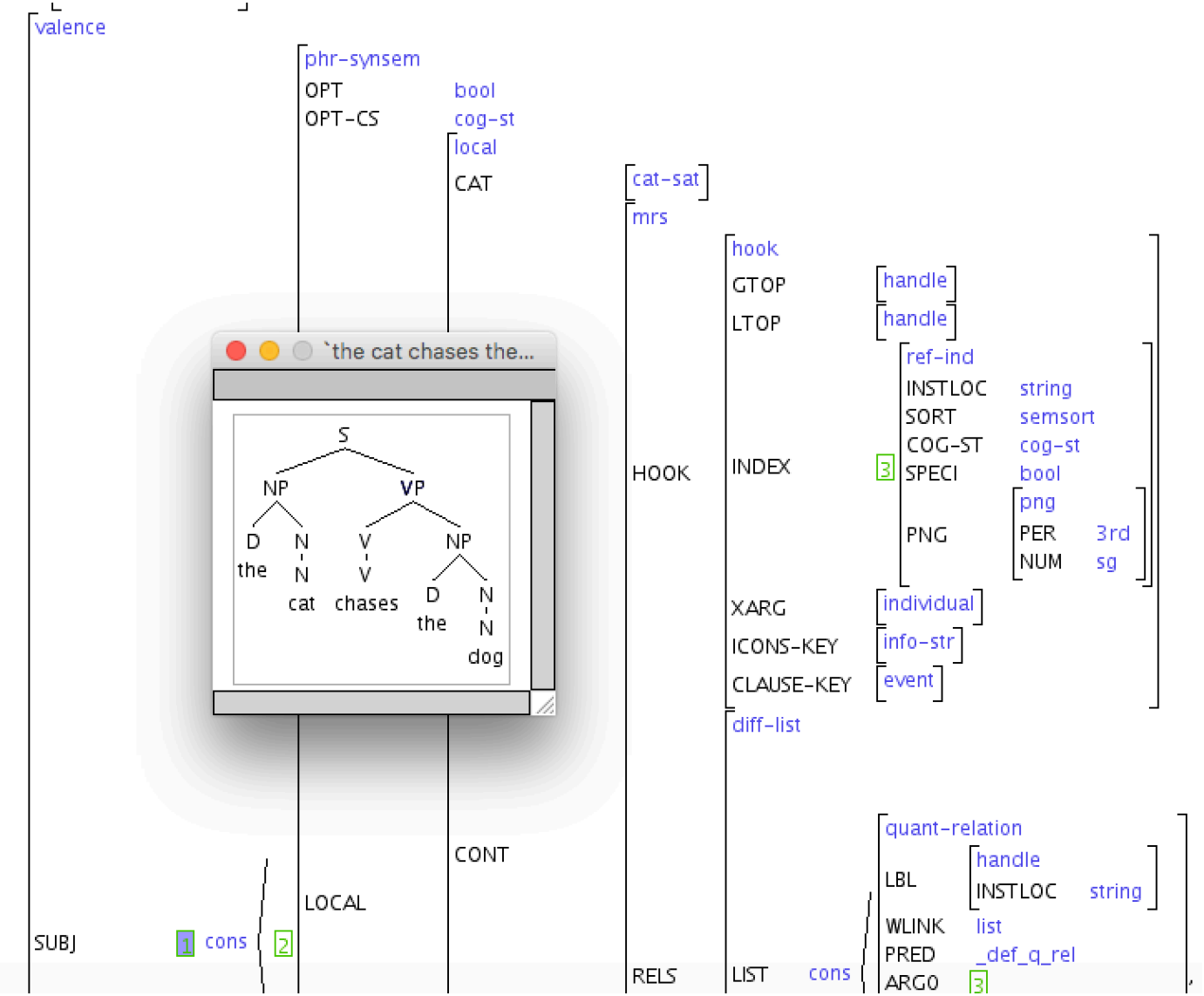
Parsing with Matrix grammars (Copestake 2002)

- ...The files are loaded into parsing software, e.g. LKB
- The grammar can now be used to parse (lists of) sentences:



HPSG (Pollard and Sag 1994)

Head-Driven Phrase Structure Grammar structures

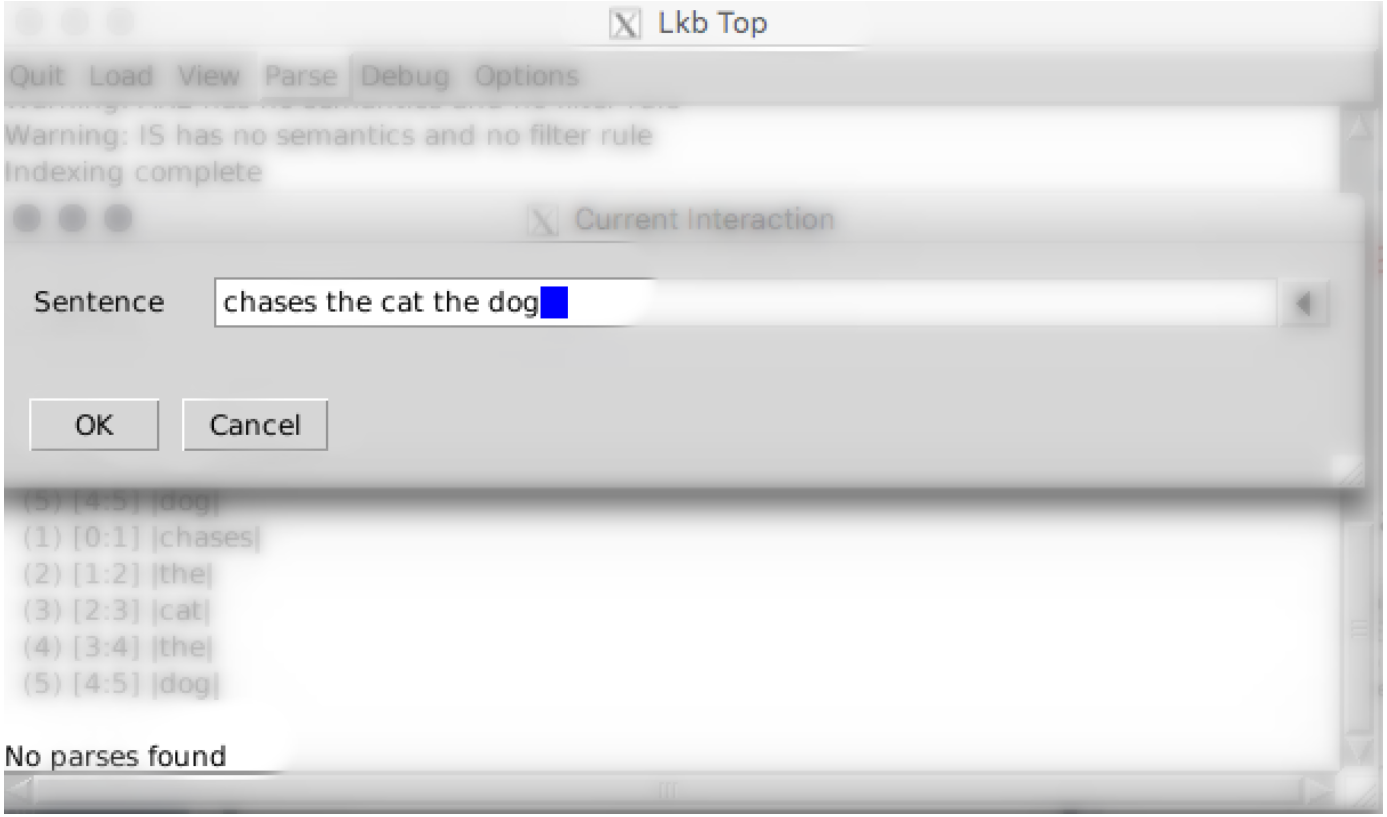


Minimal Recursion Semantics (Copestake et al. 2005)

$$
\begin{bmatrix}
\text{TOP} & h0 \\
\text{INDEX} & e2 \\
\text{RELS} & \left\langle \begin{bmatrix} \text{_the_q}\langle 0{:}3\rangle & \\ \text{LBL} & h4 \\ \text{RSTR} & h5 \\ \text{ARG0} & x3 \\ \text{BODY} & h6 \end{bmatrix}, \begin{bmatrix} \text{_cat_n_1}\langle 4{:}7\rangle & \\ \text{LBL} & h7 \\ \text{ARG0} & x3 \end{bmatrix}, \begin{bmatrix} \text{_chase_v_1}\langle 8{:}14\rangle & \\ \text{LBL} & h1 \\ \text{ARG1} & x3 \\ \text{ARG2} & x8 \\ \text{ARG0} & e2 \end{bmatrix}, \begin{bmatrix} \text{_the_q}\langle 15{:}18\rangle & \\ \text{LBL} & h9 \\ \text{RSTR} & h10 \\ \text{ARG0} & x8 \\ \text{BODY} & h11 \end{bmatrix}, \begin{bmatrix} \text{_dog_n_1}\langle 19{:}22\rangle & \\ \text{LBL} & h12 \\ \text{ARG0} & x8 \end{bmatrix} \right\rangle \\
\text{HCONS} & \left\langle \begin{bmatrix} qeq & \\ \text{HARG} & h0 \\ \text{LARG} & h1 \end{bmatrix}, \begin{bmatrix} qeq & \\ \text{HARG} & h10 \\ \text{LARG} & h12 \end{bmatrix}, \begin{bmatrix} qeq & \\ \text{HARG} & h5 \\ \text{LARG} & h7 \end{bmatrix} \right\rangle
\end{bmatrix}
$$

Ungrammatical examples not admitted

For example, the grammar we just created will not parse sentences which are not SVO:



LING567 Preview

- A grammar engineering class
- Use HPSG to build a small grammar of a language
- Used to be a language of your choice; you read the grammar, then obtain the starter grammar by filling out the Matrix questionnaire
- Now: you are handed a (somewhat noisy) starter grammar inferred automatically from a dataset (AGGREGATION)
- In any case, what you do is develop the starter grammar further
- A unique class not offered too often elsewhere
- People tend to spend many hours a week on this class but they usually do not regret it (it is an **elective** for a reason)

Open questions in grammar engineering

- Uptake
  - Why aren't more syntacticians and field linguists (and NLP researchers, for that matter) doing GE?
  - the above includes HPSG syntacticians
- Formalism issues
  - Direct correspondence to something someone believes to be the current theory (lack thereof)
  - Efficiency vs. parsimony tension
    - Fixed order and number of daughters in a rule; not "current HPSG"!
  - "Essential" vs. "incidental" complexity (Brooks 1974)?
- "Engineering debt"
  - In order to test complex claim, lots of groundwork needs to be done
  - Example from real life: had to implement adverbs and adpositions in the Matrix to start working on wh-questions

References I

Emily M. Bender, Dan Flickinger, and Stephan Oepen. 2002. The Grammar Matrix: An open-source starter-kit for the rapid development of cross-linguistically consistent broad-coverage precision grammars. In John Carroll, Nelleke Oostdijk, and Richard Sutcliffe, editors, *Proceedings of the Workshop on Grammar Engineering and Evaluation at the 19th International Conference on Computational Linguistics*, pages 8–14, Taipei, Taiwan, 2002.

Emily M. Bender, Scott Drellishak, Antske Fokkens, Laurie Poulson, and Safiyyah Saleem. 2010. Grammar customization. *Research on Language & Computation*, 8(1):23–72. ISSN 1570-7075. URL http://dx.doi.org/10.1007/s11168-010-9070-1. 10.1007/s11168-010-9070-1.

Frederick P Brooks. 1974. The mythical man-month. *Datamation*, 20(12):44–52.

Ann Copestake. 2002. *Implementing typed feature structure grammars*. CSLI publications.

Ann Copestake, Dan Flickinger, Carl Pollard, and Ivan A Sag. 2005. Minimal recursion semantics: An introduction. *Research on language and computation*, 3(2-3):281–332.

Carl Pollard and Ivan A. Sag. 1994. *Head-Driven Phrase Structure Grammar*. Studies in Contemporary Linguistics. The University of Chicago Press and CSLI Publications, Chicago, IL and Stanford, CA.