Introduction to Computational Linguistics

Parsing with Unification. Grammar engineering

Olga Zamaraeva (2018)
Based on Bender (prev. years)

University of Washington
May 15, 2018
Logistics

▶ Due Friday:
  ▶ Project, Milestone 3
  ▶ Assignment 4

▶ Assignment 5
  ▶ Can start today
  ▶ DO NOT DELAY
Overview

- Unification parsing
  - high level
- Linguistic hypotheses examples
- Grammar Engineering
- Grammar Matrix demo
- LKB demo (for Assignment 5)
Parsing with Unification

- **Input**: FS1 and FS2
- **Output**: FS3 (combining all constraints from input)
- (or failure)
- Examples:
  - What does unification look like for a unary rule?
  - What does unification look like for a ternary rule?
Unification under unary/binary rules

HeadSpecifier Rule 1:

\[
\left[\text{phrase} \left[ \begin{array}{c} \text{VAL} \\
\text{COMPS} \\
\text{SPR} \\
+ \end{array} \right] \right] \rightarrow \left[ \begin{array}{c} \text{NP} \\
\text{HEAD} \\
\text{AGR} \\
1 \end{array} \right] \\
\left[ \begin{array}{c} \text{phrase} \\
\text{HEAD} \\
\text{verb} \\
\text{AGR} \\
1 \end{array} \right] \\
\left[ \begin{array}{c} \text{VAL} \\
\text{SPR} \\
- \end{array} \right]
\]

HeadSpecifier Rule 2:

\[
\left[\text{phrase} \left[ \begin{array}{c} \text{VAL} \\
\text{COMPS} \\
\text{SPR} \\
+ \end{array} \right] \right] \rightarrow \text{D} \left[ \begin{array}{c} \text{phrase} \\
\text{HEAD} \\
\text{noun} \\
\text{VAL} \\
\text{SPR} \\
- \end{array} \right]
\]
Unification under unary-binary rules

Which rule licenses each node?

Note the three separate uses of DAGs

© 2003 CSLI Publications
Unification under unary/binary rules

- Mothers and Daughters
- A rule can be encoded as a type
  - with appropriate features: Head-Daughter and Non-head-daughter
Binary rule as a feature structure

Head-Subject Rule
- SUBJ: ⟨⟩
- COMPS: ⟨⟩
- HEAD-DTR: [SUBJ ⟨1⟩, COMPS ⟨⟩]
- NON-HEAD-DTR: 1

Head-Complement Rule
- COMPS: ⟨⟩
- SUBJ: ⟨⟩
- H-DTR: [COMPS ⟨2⟩]
- NH-DTR: 2

Root
- HSR
  - VP
    - HCR
      - [S ⟨1⟩]
      - [O ⟨2⟩]
Use graphs (DAGs) to represent feature structures
  - The model is not regular or context-free, however
  - We need to modify lists, so turing-complete
  - A limitation of DAGs: no loops
    - Difficulties with reduplication and recursive morphology

Augment these structures with another layer, adding features ‘pointer’ and ‘content’

Use pointers to merge the graphs representing the two input feature structures (why?)

Unification is recursive (why?)

Unification is destructive (why?)
Parsing with unification

\[
\begin{align*}
\text{NUMBER} \ s\text{g} & \& \text{PERSON} \ 3\text{rd}
\end{align*}
\]
Parsing with unification

\[
\begin{bmatrix}
\text{NUMBER} & sg \\
\text{PERSON} & \text{person}
\end{bmatrix} \& \begin{bmatrix}
\text{PERSON} & 3rd
\end{bmatrix}
\]
Parsing with unification
Parsing with unification

\[\text{CONTENT} \rightarrow \text{NUMBER} \rightarrow \text{PERSON} \rightarrow \text{CONTENT} \rightarrow \text{POINTER} \rightarrow \text{NULL} \rightarrow \text{CONTENT} \rightarrow \text{POINTER} \rightarrow \text{NULL} \rightarrow \text{CONTENT} \rightarrow \text{POINTER} \rightarrow \text{NULL} \rightarrow s_g \]

\[\text{CONTENT} \rightarrow \text{POINTER} \rightarrow \text{NULL} \rightarrow \text{CONTENT} \rightarrow \text{POINTER} \rightarrow \text{NULL} \rightarrow \text{CONTENT} \rightarrow \text{POINTER} \rightarrow \text{NULL} \rightarrow 3_{rd} \]

\[\text{CONTENT} \rightarrow \text{POINTER} \rightarrow \text{NULL} \rightarrow \text{CONTENT} \rightarrow \text{POINTER} \rightarrow \text{NULL} \rightarrow \text{CONTENT} \rightarrow \text{POINTER} \rightarrow \text{NULL} \rightarrow \text{NULL} \]
Parsing with unification
Parsing with unification

\[
\begin{align*}
\text{AGR} \begin{bmatrix} 1 \text{NUMBER} \ sg \end{bmatrix} & \& \text{SUBJ} \begin{bmatrix} \text{AGR} \begin{bmatrix} \text{PERSON} \ 3rd \end{bmatrix} \end{bmatrix} \\
\text{SUBJ} \begin{bmatrix} \text{AGR} \ 1 \end{bmatrix} & 
\end{align*}
\]
Grammar Engineering

- Implementing grammars in linguistically motivated syntactic and semantic formalisms
- Why do that?
Grammar Engineering

- Implementing grammars in linguistically motivated syntactic and semantic formalisms
  - Popular formalisms: LFG, HPSG
- Why do that?
  - Linguistic hypotheses testing
  - Increasing consistency in language documentation
  - What about “real NLP” applications?
Grammar Engineering

- Implementing grammars in linguistically motivated syntactic and semantic formalisms
  - Popular formalisms: LFG, HPSG
- Why do that?
  - Linguistic hypotheses testing
  - Increasing consistency in language documentation
  - What about “real NLP” applications?
    - Possible, but time-consuming
    - ERG is being developed for over 25 years now
Linguistic hypotheses testing

“...it is extremely easy to think you have a solution to a problem when in fact you don’t” (Partee, 1979)
Linguistic hypotheses testing

- Identify key examples
- Develop analysis
- Identify cases of interesting predictions
- Test acceptability of new key examples
- Refine analysis
- Identify phenomena to analyze

What’s wrong with paper-and-pencil syntax?

Linguistic hypotheses testing

- What’s wrong with paper-and-pencil syntax?
  - Easy to overlook counterexamples/inconsistencies
Linguistic hypotheses testing

- What’s wrong with paper-and-pencil syntax?
  - Easy to overlook counterexamples/inconsistencies
    - The analysis must correctly account for the data that motivated it
Linguistic hypotheses testing

- What’s wrong with paper-and-pencil syntax?
  - Easy to overlook counterexamples
    - The analysis **must** correctly account for the data that motivated it
  - How to check that an addition did not break the existing system?
Linguistic hypotheses testing

- What’s wrong with paper-and-pencil syntax?
  - Easy to overlook counterexamples
    - The analysis must correctly account for the data that motivated it
  - How to check that an addition did not break the existing system?
    - How does the analysis of a subsystem generalize?
Linguistic hypotheses testing

- What’s wrong with paper-and-pencil syntax?
  - Easy to overlook counterexamples
    - The analysis **must** correctly account for the data that motivated it
  - How to check that an addition did not break the existing system?
    - How does the analysis of a subsystem generalize?
  - Can also do inference over data
    - ML methods are also appropriate
    - But we don’t always have enough data for ML
Syntactic hypothesis

- Syntax: mapping of strings to meaning/structure; model of well-formedness
- Syntactic hypothesis: hypothesis about the structures assigned to a class of sentences and about constraints on possible grammars
Syntactic hypotheses: examples (broad)

- *Principles and parameters*: particular grammars are composed of universal principles and settings for particular parameters
- *Semantic compositionality*: the meaning of an expression is computable from the meaning of its parts
  - This is *sentence meaning!* (as opposed to speaker meaning)
- Natural languages are context free
  - disproven! (how?)
  - ...which btw means it was an excellent hypothesis! (why?)
Syntactic hypotheses: examples (narrow)

- Russian has free word order
- Chintang has affix iteration
- Piraha has no recursion
  - (How does this one refer to one broad hypothesis about language?)

Easy to test?..
Syntactic hypotheses: examples (even narrower?)

- The word *the* is very frequent in English
- Finnish L2 speakers emphasize phoneme length
- Turkish allows complementizers before and after the complement clause

Easy to test?..
Syntactic hypotheses: examples (formalism)

- Case is a HEAD feature (HPSG)
  - How is this hypothesis different from the other ones that we saw?
Hypothesis testing: examples

- Russian has free word order
  - What does it really mean?
  - Can I scramble words?
    - Need to **define** WO
    - and even then, may run into problems
    - So does it mean it does not have a free WO?
Hypothesis testing: examples

- *Turkish allows complementizers before and after the complement clause*
  - Yes, but one complementizer is used more than the other
  - One may be influenced by language contact more than the other
Hypothesis testing: examples

- “The verb perform doesn’t take mass noun complements”
  - (said Chomsky in a talk)
  - *perform magic* – says someone from the audience
  - Has Chomsky’s hypothesis been disproven?
Theory, Formalism, and Framework

- **Theory**
  - a set of major ideas and principles
  - For example?

- **Formalism**
  - A concrete way to encode the ideas and principles
  - For example?

- **Framework**
  - Either something even wider than a theory
  - Or also: a set of tools and implementations
Theory, Formalism, and Framework

- **Theory**
  - a set of major ideas and principles
  - E.g. the Head Feature Principle in HPSG or binary branching in MP

- **Formalism**
  - A concrete way to encode the ideas and principles
  - E.g. binary vs ternary branching in some HPSG-based formalisms

- **Framework**
  - Either something even wider than a theory
  - Or also: a set of tools and implementations
Grammar Engineering initiatives

- The DELPH-IN research consortium
  - DEep Language Processing with Hpsg INitiative

- Other grammar engineering efforts:
  - CoreGram (HFG)
    - What is the difference with DELPH-IN?
  - ParGram (LFG)
  - Grammar construction in the Minimalist program
    (https://search.proquest.com/docview/1868419023?pq-origsite=gscholar)
Requirements for GE

- Formalism stability
- Reasonably fast parsing
  - 15-20 years ago, running deep parsers could take whole night!
- development tools
  - Starting every grammar from scratch is tedious (and inconsistent with the whole idea)
  - Consistency in editing
  - Regression testing
GE example: The Grammar Matrix

- A GE starter toolkit
- Typological breadth + syntactic depth
- A cross-linguistic account of a number of phenomena
  - A it customization system
- Demo: http://matrix.delph-in.net/customize/matrix.cgi
DELPH-IN formalism (TDL)

agreement := *top* & [ PERSON person, NUMBER number ].

\[
\begin{bmatrix}
\text{AGREEMENT} \\
\text{NUMBER} \quad number \\
\text{PERSON} \quad person
\end{bmatrix}
\]