Assembling Syntax: Modeling Constituent Questions in a Grammar Engineering Framework
Olga Zamaraeva

Department of Linguistics, University of Washington
Seminar on the Interactions between Formal and Computational Linguistics
June 1 2021
Part I: Introduction. Formal or Computational?

▶ Common point of view:
▶ CL is a statistical field which could interact with formal linguistic theory more
▶ “computational” = “statistical”

Rafael’s School of Athens (circa 1510)
Part I: Introduction. Formal or Computational?

- Common point of view:
  - CL is a statistical field which could interact with formal linguistic theory more
  - “computational” = “statistical”
- Computational linguistics can be formal and statistical (not an “or”)

Rafael’s School of Athens (circa 1510)
Part I: Introduction. Formal or Computational?

- Common point of view:
  - CL is a statistical field which could interact with formal linguistic theory more
  - “computational” = “statistical”
- Computational linguistics can be formal and statistical (not an “or”)
- Any field can!

Rafael’s School of Athens (circa 1510)
Part I: Introduction. Formal or Computational?

- Common point of view:
  - CL is a statistical field which could interact with formal linguistic theory more
  - “computational” = “statistical”
- Computational linguistics can be formal **and** statistical (not an “or”)
- **Any field** can!
- Formal is to statistical how rationalist is to empiricist
Part I: Introduction. Formal or Computational?

Common point of view:
- CL is a statistical field which could interact with formal linguistic theory more
- “computational” = “statistical”

Computational linguistics can be formal **and** statistical (not an “or”)

Any field can!
- Formal is to statistical how rationalist is to empiricist
- Rationalist is to empiricist how Idealist is to Materialist

Rafael’s School of Athens (*circa* 1510)
Part I: Introduction. Formal or Computational?

- Common point of view:
  - CL is a statistical field which could interact with formal linguistic theory more
  - “computational” = “statistical”

- Computational linguistics can be formal and statistical (not an “or”)

- Any field can!
- Formal is to statistical how rationalist is to empiricist
- Rationalist is to empiricist how Idealist is to Materialist

- Ancient debate with no right answer
Big research question

- What is the range of variation in human languages?
Big research question

▶ What is the range of variation in human languages? 🌍
▶ Many ways to approach and many relevant fields:
Big research question

- What is the range of variation in human languages? 🌍
- Many ways to approach and many relevant fields:
  - Typology
Big research question

- What is the range of variation in human languages? 🌍
- Many ways to approach and many relevant fields:
  - Typology
  - Syntactic, semantic, phonological, acquisition, and other theories
Big research question

- What is the range of variation in human languages? 🌍
- Many ways to approach and many relevant fields:
  - Typology
  - Syntactic, semantic, phonological, acquisition, and other theories
  - Computational linguistics (formal and statistical)
Big research question

- What is the range of variation in human languages? 🌍
- Many ways to approach and many relevant fields:
  - Typology
  - Syntactic, semantic, phonological, acquisition, and other theories
  - Computational linguistics (formal and statistical)
  - Natural language processing
Big research question

- What is the range of variation in human languages? 🌍
- Many ways to approach and many relevant fields:
  - Typology
  - Syntactic, semantic, phonological, acquisition, and other theories
  - Computational linguistics (formal and statistical)
  - Natural language processing
- All of these fields are probably invested in understanding the range of variation in human language
What is the range of variation in human languages?

Many ways to approach and many relevant fields:

- Typology
- Syntactic, semantic, phonological, acquisition, and other theories
- Computational linguistics (formal and statistical)
- Natural language processing

All of these fields are probably invested in understanding the range of variation in human language

Today: Computational syntax with HPSG

- Typologically informed
Natural Language Processing and CL

- CL may mean “answering linguistic questions with computational means”
  - E.g. Typological correlations
  - E.g. Grammar coverage and overgeneration
  - E.g. Lexical class membership
Natural Language Processing and CL

- CL may mean “answering linguistic questions with computational means”
  - E.g. Typological correlations
  - E.g. Grammar coverage and overgeneration
  - E.g. Lexical class membership
- NLP (often also called CL!) may or may not target “linguistic questions”
Natural Language Processing and CL

- CL may mean “answering linguistic questions with computational means”
  - E.g. Typological correlations
  - E.g. Grammar coverage and overgeneration
  - E.g. Lexical class membership
- NLP (often also called CL!) may or may not target “linguistic questions”
  - Learn models from data
Natural Language Processing and CL

- CL may mean “answering linguistic questions with computational means”
  - E.g. Typological correlations
  - E.g. Grammar coverage and overgeneration
  - E.g. Lexical class membership
- NLP (often also called CL!) may or may not target “linguistic questions”
  - Learn models from data
  - Use models to:
Natural Language Processing and CL

- CL may mean “answering linguistic questions with computational means”
  - E.g. Typological correlations
  - E.g. Grammar coverage and overgeneration
  - E.g. Lexical class membership
- NLP (often also called CL!) may or may not target “linguistic questions”
  - Learn models from data
  - Use models to:
    - Perform language tasks
Natural Language Processing and CL

- CL may mean “answering linguistic questions with computational means”
  - E.g. Typological correlations
  - E.g. Grammar coverage and overgeneration
  - E.g. Lexical class membership

- NLP (often also called CL!) may or may not target “linguistic questions”
  - Learn models from data
  - Use models to:
    - Perform language tasks
    - Learn something about faculties involved in those tasks
Does NLP care about range of language variation?

- Linguists (including computational):
  - Finding/describing/analyzing range of variation is a **fundamental** goal
Does NLP care about range of language variation?

- Linguists (including computational):
  - Finding/describing/analyzing range of variation is a **fundamental** goal
- NLP:
Does NLP care about range of language variation?

- Linguists (including computational):
  - Finding/describing/analyzing range of variation is a **fundamental** goal
- NLP:
  - For learning about language faculties:
Does NLP care about range of language variation?

▶ Linguists (including computational):
  ▶ Finding/describing/analyzing range of variation is a **fundamental** goal
▶ NLP:
  ▶ For learning about language faculties:
    ▶ Cares similarly to how linguists do¹

---

¹ Look at any SCiL conference program
Does NLP care about range of language variation?

▶ Linguists (including computational):
  ▶ Finding/describing/analyzing range of variation is a fundamental goal
▶ NLP:
  ▶ For learning about language faculties:
    ▶ Cares similarly to how linguists do¹

¹ Look at any SCiL conference program
Does NLP care about range of language variation?

- Linguists (including computational):
  - Finding/describing/analyzing range of variation is a fundamental goal
- NLP:
  - For learning about language faculties:
    - Cares similarly to how linguists do¹
  - For performing tasks:
    - Can we just transfer from high-resource?

¹ Look at any SCiL conference program
Does NLP care about range of language variation?

- Linguists (including computational):
  - Finding/describing/analyzing range of variation is a fundamental goal
- NLP:
  - For learning about language faculties:
    - Cares similarly to how linguists do¹
  - For performing tasks:
    - Can we just transfer from high-resource?
    - From an idealist/rationalist standpoint:

¹ Look at any SCiL conference program
Does NLP care about range of language variation?

▶ Linguists (including computational):
  ▶ Finding/describing/analyzing range of variation is a **fundamental** goal

▶ NLP:
  ▶ For learning about language faculties:
    ▶ Cares similarly to how linguists do\(^1\)
  ▶ For performing tasks:
    ▶ Can we just transfer from high-resource?
    ▶ From an **idealist/rationalist** standpoint:
    ▶ Systematic knowledge is **core** to evaluating data-driven models

\(^1\) Look at any SCiL conference program
Does NLP care about range of language variation?

- Linguists (including computational):
  - Finding/describing/analyzing range of variation is a fundamental goal
- NLP:
  - For learning about language faculties:
    - Cares similarly to how linguists do\(^1\)
  - For performing tasks:
    - Can we just transfer from high-resource?
    - From an idealist/rationalist standpoint:
    - Systematic knowledge is core to evaluating data-driven models
    - ...need to reason about what we’ve done

---

\(^1\) Look at any SCiL conference program
Does NLP care about range of language variation?

▶ Linguists (including computational):
  ▶ Finding/describing/analyzing range of variation is a **fundamental** goal
▶ NLP:
  ▶ For learning about language faculties:
    ▶ Cares similarly to how linguists do\(^1\)
  ▶ For performing tasks:
    ▶ Can we just transfer from high-resource?
    ▶ From an **idealistic/rationalist** standpoint:
    ▶ Systematic knowledge is **core** to evaluating data-driven models
    ▶ ...need to **reason** about what we’ve done
    ▶ PTB, PSD, UD (which were created by people) as **target** representations

---

\(^1\) Look at any SCiL conference program
How to study languages?!

- Linguistic typology
How to study languages?! 

- Linguistic typology
  - Study range of variation wrt **broad**, diverse characteristics
  - Syntactic (semantic, morphological, phonological, discourse...) theory吸入
How to study languages?!

- Linguistic typology
  - Study range of variation wrt broad, diverse characteristics
- Syntactic (semantic, morphological, phonological, discourse...) theory
  - In-depth inquiry into modeling phenomena “elegantly”
How to study languages?!

- **Linguistic typology**
  - Study range of variation wrt **broad**, diverse characteristics
- **Syntactic (semantic, morphological, phonological, discourse...) theory**
  - **In-depth** inquiry into modeling phenomena “elegantly”
- **Computational linguistics**
  - formal (implement and test theories)
  - statistical (learn patterns bottom-up from data and generalize)
How to study languages?!

- Linguistic typology
  - Study range of variation wrt broad, diverse characteristics
- Syntactic (semantic, morphological, phonological, discourse...) theory
  - In-depth inquiry into modeling phenomena “elegantly”
- Computational linguistics
  - formal (implement and test theories)
  - statistical (learn patterns bottom-up from data and generalize)
- Traditionally separate
How to study languages?!

- Linguistic typology
  - Study range of variation wrt **broad**, diverse characteristics
- Syntactic (semantic, morphological, phonological, discourse...) theory
  - **In-depth** inquiry into modeling phenomena “elegantly”
- Computational linguistics
  - formal (implement and test theories)
  - statistical (learn patterns bottom-up from data and generalize)
- Traditionally separate
  - Can we combine them?
Combining breadth, depth, ideas, and empiricism

- **Computational modeling**
  - of the theory
  - ...for reproducibility and rigor
  - with emphasis on data patterns
  - for robustness
  - Goal: assemble fragments of in-depth analyses
  - ...into a typologically diverse system
  - ...which is fully implemented
Computational syntax

- Implement grammars on the computer 📱
Computational syntax

- Implement grammars on the computer
- Run grammars automatically on sentences
Computational syntax

- Implement grammars on the computer
- Run grammars automatically on sentences
  - ...as many sentences as you have
Computational syntax

- Implement grammars on the computer 📱
- Run grammars automatically on sentences 📝
  - ...as many sentences as you have 📖
  - ...from as many languages as you have data from
Computational syntax

- Implement grammars on the computer 🌐
- Run grammars automatically on sentences 📆
  - ...as many sentences as you have 📆
  - ...from as many languages as you have data from 🌍
  - ...including typologically diverse languages 🌍
Computational syntax

- Implement grammars on the computer 🎮
- Run grammars automatically on sentences 📝
  - ...as many sentences as you have 📝
  - ...from as many languages as you have data from
  - ...including typologically diverse languages 🌍
  - ...as many times as you need
Computational syntax

- Implement grammars on the computer 🖥️
- Run grammars automatically on sentences 📝
  - ...as many sentences as you have 📖📖📖
  - ...from as many languages as you have data from
  - ...including typologically diverse languages 🌍
  - ...as many times as you need
- Grow grammars and accumulate knowledge artifacts
Computational syntax

- Implement grammars on the computer
- Run grammars automatically on sentences
  - ...as many sentences as you have
  - ...from as many languages as you have data from
  - ...including typologically diverse languages
  - ...as many times as you need
- Grow grammars and accumulate knowledge artifacts
- Growing the area of applicability of a set of hypotheses which grammars represent
Philosophy: Method of fragments and Pragmatism

- Fully explicit **grammar fragments**\(^2\) that can be extended
  - ...constitute research artifacts that can be literally built upon
  - ...together and over time, contribute to our understanding of syntax

https://www.theinformationlab.co.uk/2017/08/09/data-scaffolding-easy-steps-fill-missing-data/

\(^2\) Montague 1974
Interim summary for Part I

- Philosophy of science applies to CL like to any field
- Empiricism and Rationalism have been at odds for centuries
- But they continue to walk hand in hand
- Next: Computational syntax
  - Implementing formal theories
Part II: Formal depth, implemented on computer

\[
\begin{align*}
\text{word} & \quad \text{ORTH} \langle \text{语法} \rangle \\
\text{SYN} | \text{CAT} | \text{SUBCAT} & \langle \text{DET} \rangle \\
\text{SEM} & \quad \begin{cases} 
\text{IND} & 0 \\
\text{RESTR} & \left\{ \begin{array}{c}
\text{grammar} \\
\text{INST} & 0
\end{array} \right. 
\end{cases}
\end{align*}
\]

\[
\begin{align*}
\text{word} & \quad \text{ORTH} \langle \text{دستور} \rangle \\
\text{SYN} | \text{CAT} | \text{SUBCAT} & \langle \text{DET} \rangle \\
\text{SEM} & \quad \begin{cases} 
\text{IND} & 0 \\
\text{RESTR} & \left\{ \begin{array}{c}
\text{grammar} \\
\text{INST} & 0
\end{array} \right. 
\end{cases}
\end{align*}
\]

https://hpsg.hu-berlin.de/hpsg2020/
Head-Driven Phrase Structure Grammar

- **Fully explicit** formalism
- Lexicalist and surface-oriented
- A grammar is a hierarchy of types encoded as feature structures where features are constrained to have some values
- A structure licensing a sentence must be well-formed

\[
\begin{align*}
\text{subj-head-phrase} & \quad \text{SUBJ} \quad \langle \rangle \\
\text{HEAD-DTR} & \quad 2 \left[ \text{SUBJ} \quad \langle 1 \rangle \right] \\
\text{NON-HEAD-DTR} & \quad 1 \\
\text{ARGS} & \quad \langle 1, 2 \rangle
\end{align*}
\]

---

3 Pollard and I. A. Sag 1994
HPSG Phrase Structure Rule

- Describes a feature structure that is a phrase and can be visualized as a tree
- “Mother” and “daughter” nodes
- Identities (tags)

```
S

[1] NP  VP
Ivan [SUBJ ⟨1⟩]
sleeps

subj-head-phrase

SUBJ ⟨⟩
HEAD-DTR 2 [SUBJ ⟨1⟩]
NON-HEAD-DTR 1
ARGS ⟨1, 2⟩
```
HPSG: Theoretical and Computational

- Both types of research exist
- HPSG formalism can be used to posit multiple theories
- DELPH-IN HPSG
  - International research consortium
  - Restricted version of HPSG formalism

---

Assembling Syntax

Olga Zamaraeva

Introduction

Computational syntax with HPSG

Assembling typologically diverse analyses

References

DelPH-IN main projects

- The English Resource Grammar (ERG)\(^5\)
  - **Broad** coverage; used in NLP\(^6\)
    - Semantic representations (ERS, MRS, DMRS) used widely for evaluating semantic parsers\(^7\)

---

5 Flickinger 2000, 2011
7 Oepen and Flickinger 2019
Assembling Syntax

Olga Zamaraeva

Introduction

Computational syntax with HPSG

Assembling typologically diverse analyses

References

DELPH-IN main projects

▶ The English Resource Grammar (ERG)\(^5\)
  ▶ Broad coverage; used in NLP\(^6\)
    ▶ Semantic representations (ERS, MRS, DMRS) used widely for evaluating semantic parsers\(^7\)

▶ Medium-size grammars of Japanese, Chinese, German, Spanish...\(^8\)

---

5 Flickinger 2000, 2011
7 Oepen and Flickinger 2019
8 Siegel et al. 2016; Fan 2018; Crysmann 2003; Marimon 2010
DELPH-IN main projects

▶ The English Resource Grammar (ERG) 5
  ▶ Broad coverage; used in NLP 6
    ▶ Semantic representations (ERS, MRS, DMRS) used widely for evaluating semantic parsers 7

▶ Medium-size grammars of Japanese, Chinese, German, Spanish... 8
▶ The Grammar Matrix: 9 Automated starter grammars; typologically-driven (Part III)

5 Flickinger 2000, 2011
7 Oepen and Flickinger 2019
8 Siegel et al. 2016; Fan 2018; Crysmann 2003; Marimon 2010
9 Bender, Flickinger, and Oepen 2002; Bender, Drellishak, et al. 2010
Assembling Syntax

Olga Zamaraeva

Introduction

Computational syntax with HPSG

Assembling typologically diverse analyses

References

DELPH-IN main projects

- The English Resource Grammar (ERG)\(^5\)
  - **Broad** coverage; used in NLP\(^6\)
    - Semantic representations (ERS, MRS, DMRS) used widely for *evaluating semantic parsers*\(^7\)

- **Medium**-size grammars of Japanese, Chinese, German, Spanish...\(^8\)
- The Grammar Matrix:\(^9\) Automated **starter** grammars; **typologically**-driven (Part III)
  - Bootstrap grammar development for more languages\(^10\)

---

5 Flickinger 2000, 2011
7 Oepen and Flickinger 2019
8 Siegel et al. 2016; Fan 2018; Crysmann 2003; Marimon 2010
9 Bender, Flickinger, and Oepen 2002; Bender, Drellishak, et al. 2010
10 Bender 2010; Crowgey 2019; Inman 2019
Example: First cross-linguistic analysis of constituent (wh) questions in DELPH-IN HPSG

▶ Classic set of questions for syntactic theory:
▶ How are question words (“wh-”) distributed?
▶ How to represent interrogative semantics?
   ▶ Quantification, scope, wh-words as question parameters of different clauses...
▶ How to model question word fronting (4)?
▶ How to model optional fronting (5)?

(4) Gde kto chto vidit?
   where who.NOM what.ACC see.3SG
   ‘Who sees what where?’[rus]

(5) Ty gde rabotaesh?
   you where work.3SG
   ‘Where do you work?’[rus]

---

11 Zamaraeva 2021
Fronting analysis in HPSG

- “Nonlocal” features are propagated up the tree\textsuperscript{12}
- Feature SLASH creates a long-distance dependency at the level of the verb
- The *filler-gap* rule discharges the dependencies

\textsuperscript{12} Pollard and I. A. Sag 1994; Ginzburg and I. Sag 2000
Extending the fragment: Multiple question fronting

- Recursive application of *filler-gap*\(^{13}\)
- Theoretical takeaways arising from computational implementation:
  - “Optional” fronting is hard!\(^{14}\)
  - New evidence from HPSG
  - Simpler analysis of multiple fronting → less simple morphological marking\(^{15}\)
  - Must a system be “elegant”?

(6) kto gde chto vidit?
who.NOM where what.ACC see.3SG
‘Who sees what where?’[rus]
Interim Summary

- HPSG is a fully-explicit syntactic formalism
- Can implement grammar fragments rigorously
- Most insights come from **assembling** the fragments into a system
  - ...intractable by hand
  - 🤖
- Next: The Grammar Matrix: A typologically driven meta-grammar engineering framework
Assembling and evaluating typologically diverse analyses
Assembling grammars systematically

- **Goal**: Make implemented linguistic grammars bigger and more accessible to broader research community
Assembling grammars systematically

- **Goal**: Make implemented linguistic grammars bigger and more accessible to broader research community
- **Method**: Meta-grammar engineering with Head-Driven Phrase Structure Grammar
Assembling grammars systematically

- **Goal**: Make implemented linguistic grammars bigger and more accessible to a broader research community
- **Method**: Meta-grammar engineering with Head-Driven Phrase Structure Grammar
- **Project**: Analysis of constituent questions for a grammar engineering system
Assembling grammars systematically

- **Goal**: Make implemented linguistic grammars bigger and more accessible to broader research community
- **Method**: Meta-grammar engineering with Head-Driven Phrase Structure Grammar
- **Project**: Analysis of constituent questions for a grammar engineering system
- **Result**:
  - New library in the system; more complex hypotheses can be tested
  - Archived test suites and analyses for several languages
  - Some takeaways regarding the interaction of different analyses
(4) Gde kto chto vidit?
where who.NOM what.ACC see.3SG
‘Who sees what where?’[rus]
Questionnaire – Analysis – Semantics

(4) Gde kto chto vidit?
where who NOM what ACC see 3SG
‘Who sees what where?’ [rus]

Choices regarding the position of question phrases

Question phrases can appear at the left edge of the sentence regardless of the position the questioned constituent would appear in (Who did you see? I know who you saw etc.).
- Only one question phrase can be fronted
- All question phrases can be fronted
- Question phrases cannot be fronted (stay in situ)
The Grammar Matrix

- Meta-grammar engineering framework\textsuperscript{16}
- Input: Typological specification, lexicon, morphological rules
- Output: Implemented HPSG grammar fragment
  - Parse and generate sentences
  - Output syntactic and \textit{semantic} representations
- Many syntactic phenomena are supported\textsuperscript{17}

\textsuperscript{16} https://matrix.ling.washington.edu/customize/matrix.cgi
\textsuperscript{17} Zamaraeva 2021; Zamaraeva, Howell, and Bender 2019; Howell and Zamaraeva 2018; Saleem 2010; Song 2014; Nielsen 2018; Drellishak and Bender 2005; Crowgey 2013; Bender and Flickinger 2005
Matrix libraries

- Specify several phenomena at the same time
- Click to **get a grammar fragment** covering all of them
- Test hypotheses **in interaction**
- Parse and **generate** data within fragment’s area of coverage
  - Large lexicons can be imported

---

18 There may be bugs
Matrix library development: Bottom up!

Questions are designed based on surveys of typological literature. Mapping of questionnaire answers to implemented grammar fragments is done with a hybrid, data-driven + formal methodology. Use theory but goal is to grow coverage and minimize overgeneration over test suite. Libraries are evaluated with held-out languages. Growing regression testing base. Language specs + test suites paired with “gold” semantic representations. Check automatically how any small change affects the all of the pairings.
Matrix library development: Bottom up!

- Questionnaires are designed based on surveys of typological literature.
Matrix library development: Bottom up!

- Questionnaires are designed based on surveys of typological literature 🌍
- Mapping of questionnaire answers to implemented grammar fragments is done with a hybrid, data-driven + formal methodology\(^{19}\) 📚
  - Use theory but goal is to grow coverage and minimize overgeneration over test suite

---

\(^{19}\) Bender, Drellishak, et al. 2010; Oepen and Flickinger 1998
▶ Questionnaires are designed based on surveys of typological literature.
▶ Mapping of questionnaire answers to implemented grammar fragments is done with a hybrid, data-driven + formal methodology.
   ▶ Use theory but goal is to grow coverage and minimize overgeneration over test suite.

---

Matrix library development: Bottom up!

- Questionnaires are designed based on **surveys of typological literature**.
- Mapping of questionnaire answers to implemented grammar fragments is done with a hybrid, **data-driven + formal** methodology\(^{19}\).
  - Use theory but goal is to grow coverage and minimize overgeneration over **test suite**.
- Libraries are evaluated with **held-out languages**.

---

\(^{19}\) Bender, Drellishak, et al. 2010; Oepen and Flickinger 1998
Questionnaires are designed based on surveys of typological literature.

Mapping of questionnaire answers to implemented grammar fragments is done with a hybrid, data-driven + formal methodology.

- Use theory but goal is to grow coverage and minimize overgeneration over test suite.

Libraries are evaluated with held-out languages.

Growing regression testing base:

- Language specs + test suites paired with “gold” semantic representations.
- Check automatically how any small change affects the all of the pairings.
**RQ:** What constitutes a model of a range of typologically attested ways of forming constituent questions within the given framework?

**Evaluation:** How well does the analysis generalize to a set of randomly picked “held-out” languages?
Constituent questions in the Matrix

- New library\textsuperscript{20}
- Typological scope.\textsuperscript{21}
  - Position of question phrase
    - Fronting, \textit{in situ}
    - Fronting optionality
  - Morphological marking
  - Question particles
    - position
    - obligatoriness
  - Question words

\begin{equation}
\begin{array}{c}
\text{S} \\
\text{ADV} & | & \text{S} \\
\text{gde} & | & \text{NP} \\
\text{where} & | & \text{S} \\
\text{kto} & | & \text{NP} \\
\text{who} & | & \text{S} \\
\text{chto} & | & \text{VP} \\
\text{what} & | & \text{VP} \\
\text{vidit} & | & \text{V} \\
\text{sees} & & \\
\end{array}
\end{equation}

\text{(4) Gde kto chto vidit?}
\text{where who.NOM what.ACC see.3SG}
\text{‘Who sees what where?’[rus]}

---

\text{18} Zamaraeva 2021
\text{19} König and Siemund 2007; Hagège 2008; Miyagawa 1987, \textit{inter alia}
Assembling fronting

<table>
<thead>
<tr>
<th>Specification</th>
<th>Types</th>
<th>Core?</th>
<th>New?</th>
<th>Custom features</th>
</tr>
</thead>
<tbody>
<tr>
<td>single oblig. front.</td>
<td><em>wh-ques-phrase</em></td>
<td>no</td>
<td>yes</td>
<td>SLASH</td>
</tr>
<tr>
<td></td>
<td><em>subj-, obj-, adj-ex.</em></td>
<td>no</td>
<td>no</td>
<td>none</td>
</tr>
<tr>
<td>multi oblig. front.</td>
<td><em>wh-ques-phrase</em></td>
<td>no</td>
<td>yes</td>
<td>HDR</td>
</tr>
<tr>
<td></td>
<td><em>subj-, obj-, adj-ex.</em></td>
<td>no</td>
<td>no</td>
<td>MODIFIED hasmod</td>
</tr>
<tr>
<td>single opt. front.</td>
<td><em>wh-ques-phrase</em></td>
<td>no</td>
<td>yes</td>
<td>SLASH</td>
</tr>
<tr>
<td></td>
<td><em>in-sutu-phrase</em></td>
<td>no</td>
<td>yes</td>
<td>none</td>
</tr>
<tr>
<td></td>
<td><em>subj-, obj-, adj-ex.</em></td>
<td>no</td>
<td>no</td>
<td>none</td>
</tr>
<tr>
<td>multi opt. front.</td>
<td><em>wh-ques-phrase</em></td>
<td>no</td>
<td>yes</td>
<td>MODIFIED hasmod</td>
</tr>
<tr>
<td></td>
<td><em>in-sutu-phrase</em></td>
<td>no</td>
<td>yes</td>
<td>HDR</td>
</tr>
<tr>
<td></td>
<td><em>subj-, obj-, adj-ex.</em></td>
<td>no</td>
<td>no</td>
<td>HDR</td>
</tr>
<tr>
<td>in situ (no front.)</td>
<td><em>in-sutu-phrase</em></td>
<td>no</td>
<td>yes</td>
<td>none</td>
</tr>
</tbody>
</table>

The position of question phrases customization summary

---

Zamaraeva 2021
Evaluation

▶ “Freeze” the analyses and the development
Evaluation

- “Freeze” the analyses and the development
- Grab reference grammars from “unseen” language families
Evaluation

- “Freeze” the analyses and the development
- Grab reference grammars from “unseen” language families
- Grab all examples of constituent questions from those books
Evaluation

- “Freeze” the analyses and the development
- Grab reference grammars from “unseen” language families
- Grab all examples of constituent questions from those books
- Fill out the questionnaire so as to cover the examples
Evaluation

► “Freeze” the analyses and the development
► Grab reference grammars from “unseen” language families
► Grab all examples of constituent questions from those books
► Fill out the questionnaire so as to cover the examples
► Click “create grammar”, run the grammar on the examples
Evaluation

▶ “Freeze” the analyses and the development
▶ Grab reference grammars from “unseen” language families
▶ Grab all examples of constituent questions from those books
▶ Fill out the questionnaire so as to cover the examples
▶ Click “create grammar”, run the grammar on the examples
▶ Coverage: % of examples the grammar actually covered
▶ Overgeneration % of ungrammatical examples parsed
  ▶ covered = gave correct semantic representation
Evaluation

- “Freeze” the analyses and the development
- Grab reference grammars from “unseen” language families
- Grab all examples of constituent questions from those books
- Fill out the questionnaire so as to cover the examples
- Click “create grammar”, run the grammar on the examples
- **Coverage**: % of examples the grammar actually covered
- **Overgeneration** % of ungrammatical examples parsed
  - covered = gave correct semantic representation
- How well can the system handle examples from an “unseen” language, as it is described in the reference grammar?
Evaluating the constituent questions library\textsuperscript{23}

<table>
<thead>
<tr>
<th>Language</th>
<th>ISO-639-3</th>
<th>Family</th>
<th>Gram. items</th>
<th>Coverage%</th>
<th>Question typology</th>
</tr>
</thead>
<tbody>
<tr>
<td>Apinajé</td>
<td>[apn]</td>
<td>Macro-Jê</td>
<td>17</td>
<td>82.3</td>
<td>single front.</td>
</tr>
<tr>
<td>Makah</td>
<td>[myh]</td>
<td>Wakashan</td>
<td>14</td>
<td>78.5</td>
<td>morphological, int. verbs</td>
</tr>
<tr>
<td>Pacoh</td>
<td>[pac]</td>
<td>Austroasiatic</td>
<td>26</td>
<td>84.6</td>
<td>single opt. front.</td>
</tr>
<tr>
<td>Paresi-Haliti</td>
<td>[pab]</td>
<td>Arawakan</td>
<td>64</td>
<td>56.0</td>
<td>single front., int. verbs</td>
</tr>
<tr>
<td>Jalkunan</td>
<td>[bxl]</td>
<td>Mande</td>
<td>33</td>
<td>78.8</td>
<td>in situ, particle, int. verbs</td>
</tr>
</tbody>
</table>

✓ Single fronting, particles, morphological marking, interrogative verbs

✗ Question words as predicates

💡 Grammars made correct predictions where I didn’t expect them to!

ديد Bugs consistent with theoretical predictions

---

\textsuperscript{23} Zamaraeva 2021
Conclusion

- Formal approaches to syntax are an important part of linguistics and NLP
Conclusion

- Formal approaches to syntax are an important part of linguistics and NLP
- Fully explicit formalisms like HPSG allow us to implement grammars on the computer and rigorously test them
Conclusion

- Formal approaches to syntax are an important part of linguistics and NLP
- Fully explicit formalisms like HPSG allow us to implement grammars on the computer and rigorously test them
- **Example:** An analysis of constituent questions integrated into the Grammar Matrix system
  - Clear area of applicability as archived in the specifications, test suites, and the version of the Matrix system
Conclusion

- Formal approaches to syntax are an important part of linguistics and NLP
- Fully explicit formalisms like HPSG allow us to implement grammars on the computer and rigorously test them
- **Example:** An analysis of constituent questions integrated into the Grammar Matrix system
  - Clear area of applicability as archived in the specifications, test suites, and the version of the Matrix system
- **Future:** Robust NLP evaluation and precision in NLP applications for more languages
Conclusion

- Formal approaches to syntax are an important part of linguistics and NLP
- Fully explicit formalisms like HPSG allow us to implement grammars on the computer and rigorously test them
- **Example:** An analysis of constituent questions integrated into the Grammar Matrix system
  - Clear area of applicability as archived in the specifications, test suites, and the version of the Matrix system
- **Future:** Robust NLP evaluation and precision in NLP applications for more languages
- **Future:** Encoded and tested sets of hypotheses for more languages
Conclusion

- Formal approaches to syntax are an important part of linguistics and NLP
- Fully explicit formalisms like HPSG allow us to implement grammars on the computer and rigorously test them
- **Example:** An analysis of constituent questions integrated into the Grammar Matrix system
  - Clear area of applicability as archived in the specifications, test suites, and the version of the Matrix system
- **Future:** Robust NLP evaluation and precision in NLP applications for more languages
- **Future:** Encoded and tested sets of hypotheses for more languages
- **Assembling fragments of our understanding of language**
Acknowledgments

- The work presented today was partially funded by the United States National Science Foundation under Grant No. BCS-1561833 (PI Bender).
- This work would not be possible without the DELPH-IN community.
References


Buys, Jan and Phil Blunsom (2017). "Robust Incremental Neural Semantic Graph Parsing". In: *ACL* (1).


References


References III


