Assembling Syntax: Modeling Constituent Questions in a Grammar Engineering Framework

Olga Zamaraeva

Department of Linguistics, University of Washington
SigTyp Lecture Series
May 14 2021
Range of language variation

- There are over 7000 languages in the world¹
- 90% of people speak about 10% of them
  - Many languages are being spoken less and less and may disappear

¹Language Files 12 ed.

http://chartsbin.com/view/1339
Understanding the range of language variation is important:

- For culture and society
- For science
  - One of the fundamental goals of linguistics
  - What about natural language processing?

http://chartsbin.com/view/1339
Natural Language Processing
Learn models from data
Natural Language Processing

- Learn models from data
- Use models to:
Natural Language Processing

- Learn models from data
- Use models to:
  - Perform language tasks
Natural Language Processing

- Learn models from data
- Use models to:
  - Perform language tasks
  - Learn something about faculties involved in those tasks
Learn models from data

- Use models to:
  - Perform language tasks
  - Learn something about faculties involved in those tasks
  - “Learn something about the world”\(^2\) through analyzing language data

---

\(^2\) Mark Yatskar; p.c.
Natural Language Processing

- Learn models from data
- Use models to:
  - Perform language tasks
  - Learn something about faculties involved in those tasks
  - “Learn something about the world”\(^2\) through analyzing language data

---

\(^2\) Mark Yatskar; p.c.
Natural Language Processing

- Learn models from data
- Use models to:
  - Perform language tasks
  - Learn something about faculties involved in those tasks
  - “Learn something about the world”\(^2\) through analyzing language data
    - Knowledge bases?

---

\(^2\) Mark Yatskar; p.c.
Does NLP care about range of language variation?

- Linguists:

  ▶ Natively care: Finding/describing/analyzing range of variation is a fundamental goal

  ▶ For learning about language faculties:

    ▶ Probably care, similar to like linguists do

    ▶ For performing tasks:

      ▶ Maybe we care, but not if we can successfully transfer models from a few high-resource languages?

      ▶ Sure, but how do we evaluate the performance?
Does NLP care about range of language variation?

- Linguists:
  - Naturally care: Finding/describing/analyzing range of variation is a fundamental goal
Does NLP care about range of language variation?

- Linguists:
  - Naturally care: Finding/describing/analyzing range of variation is a fundamental goal

- NLP:
Does NLP care about range of language variation?

- Linguists:
  - Naturally care: 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:
  ▶ Naturally care: Finding/describing/analyzing range of variation is a fundamental goal

▶ NLP:
  ▶ For learning about language faculties:
    ▶ Probably care, similar to like linguists do
Does NLP care about range of language variation?

- **Linguists:**
  - Naturally care: Finding/describing/analyzing range of variation is a fundamental goal

- **NLP:**
  - For learning about language faculties:
    - Probably care, similar to like linguists do
  - For performing tasks:
Does NLP care about range of language variation?

- **Linguists:**
  - Naturally care: Finding/describing/analyzing range of variation is a fundamental goal

- **NLP:**
  - For learning about language faculties:
    - Probably care, similar to like linguists do
  - For performing tasks:
    - Maybe we care, but not if we can successfully transfer models from a few high-resource languages?
Does NLP care about range of language variation?

- **Linguists:**
  - Naturally care: Finding/describing/analyzing range of variation is a fundamental goal

- **NLP:**
  - For learning about language faculties:
    - Probably care, similar to like linguists do
  - For performing tasks:
    - Maybe we care, but not if we can successfully transfer models from a few high-resource languages?
    - Sure, but how do we **evaluate** the performance?
Understanding the range of language variation is important:

- For culture and society
- For science
  - One of the **fundamental goals of linguistics**
  - Core to evaluating NLP models
  - Can transfer a model from English to another language but still need systematic knowledge about that other language to evaluate

http://chartsbin.com/view/1339
We need to study more languages...

- To learn more about human language, via linguistics or via NLP
  - ...just like other sciences do in their domains

\[Pongo\ tapanuensis.\ \text{Pic credit: Tim Laman}\]

\[Nater\ et\ al.\ 2017\]
We need to study more languages...

▶ Studying a small set of languages is also valuable! (In both linguistics and NLP!)
▶ ...but it biases the questions we ask

Pic credit: https://englishlive.ef.com/blog/language-lab/question-words/

---

Idiatov 2007; Bybee 1985
How to study languages?!

- Linguistic typology
How to study languages?!

- Linguistic typology
  - Study range of variation wrt broad characteristics
How to study languages?!

- Linguistic typology
  - Study range of variation wrt broad characteristics
  - Pay special attention to diversity
- Syntactic (semantic, morphological, phonological, discourse...) theory
How to study languages?!

▶ Linguistic typology
  ▶ Study range of variation wrt broad characteristics
  ▶ Pay special attention to diversity
▶ Syntactic (semantic, morphological, phonological, discourse...) theory
  ▶ In-depth inquiry into modeling phenomena with generative power
How to study languages?!

- **Linguistic typology**
  - Study range of variation wrt broad characteristics
  - Pay special attention to diversity

- **Syntactic (semantic, morphological, phonological, discourse...) theory**
  - In-depth inquiry into modeling phenomena with generative power
    - Have a model of e.g. syntax s.t. the grammar generates only correct sentences
How to study languages?!

- Linguistic typology
  - Study range of variation wrt broad characteristics
  - Pay special attention to diversity

- Syntactic (semantic, morphological, phonological, discourse...) theory
  - In-depth inquiry into modeling phenomena with generative power
    - Have a model of e.g. syntax s.t. the grammar generates only correct sentences

- Traditionally separate
How to study languages?!

- **Linguistic typology**
  - Study range of variation wrt broad characteristics
  - Pay special attention to diversity

- **Syntactic (semantic, morphological, phonological, discourse...) theory**
  - In-depth inquiry into modeling phenomena with generative power
    - Have a model of e.g. syntax s.t. the grammar generates only correct sentences

- **Traditionally separate**
  - How do we combine them?
Combining breadth and depth

- Linguistic typology
  - Study range of variation wrt broad characteristics
  - Pay special attention to diversity
- Syntactic (semantic, morphological, phonological, discourse...) theory
  - In-depth inquiry into modeling phenomena with generative power
    - Have a model of e.g. syntax s.t. the grammar generates only correct sentences
Combining breadth and depth

- Linguistic typology
  - Study range of variation wrt broad characteristics
  - Pay special attention to diversity

- Syntactic (semantic, morphological, phonological, discourse...) theory
  - In-depth inquiry into modeling phenomena with generative power
    - Have a model of e.g. syntax s.t. the grammar generates only correct sentences

- Computational modeling
  - of the theory
  - ...for reproducibility and rigor
  - ...when assembling fragments of in-depth analyses
  - ...into a typologically diverse system
Assembling Syntax

Olga Zamaraeva

Introduction

Computational syntax with HPSG

Assembling typologically diverse analyses

Future

References

- 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 🌍

Assembling Syntax
Olga Zamaraeva

Introduction

Computational syntax with HPSG

Assembling typologically diverse analyses

Future

References
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
e  - ...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

- Fully explicit grammar fragments\(^5\) 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/
Wrapping up the big picture

▶ Big Question: What is the range of variation in human languages? 🌍
Wrapping up the big picture

▶ Big Question: What is the range of variation in human languages? 🌍
Wrapping up the big picture

- Big Question: What is the range of variation in human languages? 🌍
- Many ways to approach and many relevant fields:
Wrapping up the big picture

- Big Question: What is the range of variation in human languages? 🌍
- Many ways to approach and many relevant fields:
  - Typology
Wrapping up the big picture

- Big 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
Wrapping up the big picture

- Big 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

Future
References
Wrapping up the big picture

- Big 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
  - Natural language processing
Wrapping up the big picture

- Big 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
  - Natural language processing
- All of these fields are probably invested in understanding the range of variation in human language
Wrapping up the big picture

▶ Big 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
  ▶ Natural language processing
▶ All of these fields are probably invested in understanding the range of variation in human language
  ▶ Next: Combining syntactic theory and typology
Assembling Syntax: Part II

Computational syntax with Head-Driven Phrase Structure Grammar formalism
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 demonstrably represent 📖
What is a grammar?

- Definition 1:
  - A set of rules which can be loaded into a parser to parse/generate sentences
What is a grammar?

- Definition 1:
  - A set of rules which can be loaded into a parser to parse/generate sentences

- Definition 2:
  - A set of rules which can be loaded into a parser to parse/generate sentences...
  - ...such that it gives insight into human linguistic faculties
What is a grammar?

- **Definition 1:**
  - A set of rules which can be loaded into a parser to parse/generate sentences

- **Definition 2:**
  - A set of rules which can be loaded into a parser to parse/generate sentences...
  - ...such that it gives **insight** into human linguistic faculties

- I am working somewhere in between...
Why talk about grammars to an NLP audience?

- Grammars aren’t used very much in contemporary NLP
Why talk about grammars 📖 to an NLP audience?

- Grammars 📖 aren’t used very much in contemporary ☑️ NLP
  - ...or are they? 👍
Why talk about grammars to an NLP audience?

- Grammars aren’t used very much in contemporary NLP
  - ...or are they?
- Evaluation relies on annotation
Why talk about grammars 📚 to an NLP audience?

- Grammars 📚 aren’t used very much in contemporary NLP
  - ...or are they? 🤔
- Evaluation relies on annotation 📚
- Many types of annotation rely on systematized knowledge 📚
Why talk about grammars to an NLP audience?

- Grammars aren’t used very much in contemporary NLP
  - ...or are they?
- Evaluation relies on annotation
- Many types of annotation rely on systematized knowledge
  - Grammars are a type of systematized knowledge
Why talk about grammars to an NLP audience?

- Grammars aren’t used very much in contemporary NLP
  - …or are they? 💁
- Evaluation relies on annotation 🕵️
- Many types of annotation rely on systematized knowledge 🎧
  - Grammars are a type of systematized knowledge
  - e.g. PennTree bank or UD
Why talk about grammars to an NLP audience?

- Grammars aren’t used very much in contemporary NLP
  - ...or are they?
- Evaluation relies on annotation
- Many types of annotation rely on systematized knowledge
  - Grammars are a type of systematized knowledge
  - e.g. PennTree bank or UD
  - ...someone had to systematize, formalize, and record the knowledge first
Why talk about grammars 📘 to an NLP audience?

▷ Grammars 📘 aren’t used very much in contemporary ◇ NLP
  ▷ ...or are they? 🕹️

▷ Evaluation relies on annotation 🧔

▷ Many types of annotation rely on systematized knowledge 🧩
  ▷ Grammars are a type of systematized knowledge
  ▷ e.g. PennTree bank or UD
  ▷ ...someone 🧑‍💻 had to systematize, formalize, and record the knowledge **first**
  ▷ Hard to update/improve and does not scale
Why talk about grammars to an NLP audience?

- Grammars aren’t used very much in contemporary NLP
  - ...or are they?

- Evaluation relies on annotation

- Many types of annotation rely on systematized knowledge
  - Grammars are a type of systematized knowledge
  - e.g. PennTree bank or UD
  - ...someone had to systematize, formalize, and record the knowledge first
  - Hard to update/improve and does not scale
  - ...unless you use an implemented grammar
Why talk about grammars to an NLP audience?

- Grammars aren’t used very much in contemporary NLP
  - ...or are they? 😃
- Evaluation relies on annotation 🧑
- Many types of annotation rely on systematized knowledge 📝
  -Grammars are a type of systematized knowledge
  - e.g. PennTree bank or UD
  - ...someone had to systematize, formalize, and record the knowledge first
  - Hard to update/improve and does not scale
  - ...unless you use an implemented grammar
  - **Goal:** Do this for more languages and in a more systematic and easily update-able way
Why talk about grammars to an NLP audience?

- Grammars aren’t used very much in contemporary NLP
  - ...or are they?
- Evaluation relies on annotation
- Many types of annotation rely on systematized knowledge
  - Grammars are a type of systematized knowledge
  - e.g. PennTree bank or UD
  - ...someone had to systematize, formalize, and record the knowledge first
  - Hard to update/improve and does not scale
  - ...unless you use an implemented grammar
- Goal: Do this for more languages and in a more systematic and easily update-able way
- Rely on indispensable human knowledge but support it with computational framework
Assembling grammars systematically

- Pen-and-paper syntax:
  - Consider sentences (usually a few), hypothesize an analysis
  - Test in your head
  - Revise if you notice a problem
Assembling grammars systematically

- Pen-and-paper syntax:
  - Consider sentences (usually a few), hypothesize an analysis
  - Test in your head
  - Revise if you notice a problem

- Computational syntax:
Assembling grammars systematically

- Pen-and-paper syntax:
  - Consider sentences (usually a few), hypothesize an analysis
  - Test in your head
  - Revise if you notice a problem

- Computational syntax:
  - Consider sentences (usually a few), hypothesize an analysis
Assembling grammars systematically

- Pen-and-paper syntax:
  - Consider sentences (usually a few), hypothesize an analysis
  - Test in your head
  - Revise if you notice a problem

- Computational syntax:
  - Consider sentences (usually a few), hypothesize an analysis
  - Implement the analysis in a grammar engineering framework
Assembling grammars systematically

- **Pen-and-paper syntax:**
  - Consider sentences (usually a few), hypothesize an analysis
  - Test in your head
  - Revise if you notice a problem

- **Computational syntax:**
  - Consider sentences (usually a few), hypothesize an analysis
  - Implement the analysis in a grammar engineering framework
  - Test on many sentences in literal interaction with analyses for other phenomena
Assembling grammars systematically

- Pen-and-paper syntax:
  - Consider sentences (usually a few), hypothesize an analysis
  - Test in your head
  - Revise if you notice a problem

- Computational syntax:
  - Consider sentences (usually a few), hypothesize an analysis
  - Implement the analysis in a grammar engineering framework
  - Test on many sentences in literal interaction with analyses for other phenomena
  - Revise hypotheses
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 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
Head-Driven Phrase Structure Grammar

- **Fully explicit** formalism\(^6\)
- 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**

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

\(^6\) 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

---

The English Resource Grammar (ERG) 8
  ▶ Broad coverage; used in NLP 9
  ▶ Semantic representations (ERS, MRS, DMRS) used widely for evaluating semantic parsers 10

---

8 Flickinger 2000, 2011
9 Hajdik et al. 2019; Zamaraeva, Howell, and Rhine 2018; Buys and Blunsom 2017; Packard 2014
10 Oepen and Flickinger 2019
Assembling Syntax
Olga Zamaraeva

Introduction
Computational syntax with HPSG
Assembling
typologically
diverse analyses
Future
References

DELPH-IN main projects

▶ The English Resource Grammar (ERG) 8
  ▶ Broad coverage; used in NLP 9
    ▶ Semantic representations (ERS, MRS, DMRS) used widely for evaluating semantic parsers 10

▶ Medium-size grammars of Japanese, Chinese, German, Spanish... 11

8 Flickinger 2000, 2011
9 Hajdik et al. 2019; Zamaraeva, Howell, and Rhine 2018; Buys and Blunsom 2017; Packard 2014
10 Oepen and Flickinger 2019
11 Siegel et al. 2016; Fan 2018; Crysmann 2003; Marimon 2010
Assembling Syntax

Olga Zamaraeva

Introduction
Computational syntax with HPSG
Assembling typologically diverse analyses
Future
References

DELPH-IN main projects

- The English Resource Grammar (ERG)\textsuperscript{8}
  - Broad coverage; used in NLP\textsuperscript{9}
    - Semantic representations (ERS, MRS, DMRS) used widely for evaluating semantic parsers\textsuperscript{10}

- Medium-size grammars of Japanese, Chinese, German, Spanish...\textsuperscript{11}
- The Grammar Matrix:\textsuperscript{12} Automated starter grammars; typologically-driven (Part III)

---
\textsuperscript{8} Flickinger 2000, 2011
\textsuperscript{9} Hajdik et al. 2019; Zamaraeva, Howell, and Rhine 2018; Buys and Blunsom 2017; Packard 2014
\textsuperscript{10} Oepen and Flickinger 2019
\textsuperscript{11} Siegel et al. 2016; Fan 2018; Crysmann 2003; Marimon 2010
\textsuperscript{12} Bender, Flickinger, and Oepen 2002; Bender, Drellishak, et al. 2010
DELPH-IN main projects

▶ The English Resource Grammar (ERG) ⁸
  ▶ Broad coverage; used in NLP⁹
    ▶ Semantic representations (ERS, MRS, DMRS) used widely for evaluating semantic parsers¹⁰

▶ Medium-size grammars of Japanese, Chinese, German, Spanish...¹¹
▶ The Grammar Matrix:¹² Automated starter grammars; typologically-driven (Part III)
  ▶ Bootstrap grammar development for more languages¹³

---

⁸ Flickinger 2000, 2011
⁹ Hajdik et al. 2019; Zamaraeva, Howell, and Rhine 2018; Buys and Blunsom 2017; Packard 2014
¹⁰ Oepen and Flickinger 2019
¹¹ Siegel et al. 2016; Fan 2018; Crysmann 2003; Marimon 2010
¹² Bender, Flickinger, and Oepen 2002; Bender, Drellishak, et al. 2010
¹³ Bender 2010; Crowgey 2019; Inman 2019
An 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)?
An 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)?

◼ Previously:
  ◼ Nobody had put forth an analysis of multiple fronting as in Slavic languages
  ◼ Nobody had offered a unified HPSG account of wh-questions in typologically different languages
  ◼ ...and tested it rigorously with a parser
  ◼ 14

(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]
Questionnaire – Analysis – Semantics

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

- "Nonlocal" features are propagated up the tree\(^{15}\)
- Feature \textit{SLASH} creates a long-distance dependency at the level of the verb
- The \textit{filler-gap} rule discharges the dependencies

\[ S_{\text{filler-gap}} \]

\[ [\text{SLASH } \langle \rangle] \]

\[
\text{NP} \quad \text{V} \\
\text{who} \quad [\text{SLASH } \langle 1 \rangle] \\
\text{V} \\
\text{do you think} \quad S_{\text{subj-extr}} \\
\text{[SLASH } \langle 1 \rangle] \\
\text{SUBJ } \langle \rangle \\
\text{VP} \\
\text{sleeps?} \\
\text{[SUBJ } \langle 1 \rangle] \\
\text{SLEEP } \langle 1 \rangle \\
\text{\textcolor{red}{SLASH}} \\
\text{\textcolor{red}{[\langle \rangle]}} \\
\text{\textcolor{red}{HEAD-DTR}} \\
\text{\textcolor{red}{[\langle 1 \rangle]}} \\
\text{\textcolor{red}{NON-HEAD-DTR}} \\
\text{1} \]

(5) Who do you think sleeps? [eng]

\(^{15}\) Pollard and I. A. Sag 1994; Ginzburg and I. Sag 2000
Extending the fragment: Multiple question fronting

- Allow multiple extraction and recursive application of *filler-gap*\(^{16}\)
- Takeaways:
  - “Optional” fronting is hard!\(^{17}\)
  - Simpler analysis of multiple fronting → less simple morphological marking\(^{18}\)

\[\text{filler-gap-phrase} \quad \frac{\text{SLASH}}{1} \quad \frac{\text{ARGS}}{\left[2 \left( \text{SLASH} \langle 2 \rangle \oplus 1 \right) \right]}\]

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

---

16 Zamaraeva and Emerson 2020; Crysmann 2015
17 Zamaraeva 2021
18 Zamaraeva to appear
An analysis of Slavic fronting in HPSG

Open questions:
An analysis of Slavic fronting in HPSG

► Open questions:
  ► Modeling **optional** fronting leads to **spurious** ambiguity
  ► How to get rid of it while still accounting for data?
An analysis of Slavic fronting in HPSG

Open questions:
- Modeling **optional** fronting leads to **spurious** ambiguity
  - How to get rid of it while still accounting for data?
  - **Direction**: information-structure
  - **Gist**: There is no optional fronting!
  - Nobody has done this in DELPH-IN HPSG yet
An analysis of Slavic fronting in HPSG

- Open questions:
  - Modeling **optional** fronting leads to **spurious** ambiguity
    - How to get rid of it while still accounting for data?
    - **Direction:** information-structure
    - **Gist:** There is **no** optional fronting!
    - Nobody has done this in DELPH-IN HPSG yet
  - But isn’t this a problem with **DELPH-IN HPSG** rather than with HPSG or syntactic theory in general?
An analysis of Slavic fronting in HPSG

- Open questions:
  - Modeling **optional** fronting leads to **spurious** ambiguity
    - How to get rid of it while still accounting for data?
    - **Direction:** information-structure
    - **Gist:** There is **no** optional fronting!
    - Nobody has done this in DELPH-IN HPSG yet
  - But isn’t this a problem with **DELPH-IN HPSG** rather than with HPSG or syntactic theory in general?
  - **Essential** vs. **incidental** complexity in formalisms
An analysis of Slavic fronting in HPSG

Open questions:

- Modeling **optional** fronting leads to **spurious** ambiguity
  - How to get rid of it while still accounting for data?
  - **Direction**: information-structure
  - **Gist**: There is **no** optional fronting!
  - Nobody has done this in DELPH-IN HPSG yet

- But isn’t this a problem with **DELPH-IN HPSG** rather than with HPSG or syntactic theory in general?
- **Essential** vs. **incidental** complexity in formalisms
- **Tensions in elegance** in a **typologically diverse** system
An analysis of Slavic fronting in HPSG

Open questions:

- Modeling **optional** fronting leads to **spurious** ambiguity
  - How to get rid of it while still accounting for data?
  - **Direction**: information-structure
  - **Gist**: There is no optional fronting!
  - Nobody has done this in DELPH-IN HPSG yet

- But isn't this a problem with **DELPH-IN HPSG** rather than with HPSG or syntactic theory in general?
- **Essential** vs. **incidental** complexity in formalisms
- **Tensions in elegance** in a **typologically diverse** system
- Can we have one framework for everything? 😐
An analysis of Slavic fronting in HPSG

- Open questions:
  - Modeling **optional** fronting leads to **spurious** ambiguity
    - How to get rid of it while still accounting for data?
  - **Direction:** information-structure
  - **Gist:** There is **no** optional fronting!
  - Nobody has done this in DELPH-IN HPSG yet
  - But isn’t this a problem with **DELPH-IN HPSG** rather than with HPSG or syntactic theory in general?
  - **Essential** vs. **incidental** complexity in formalisms
  - **Tensions in elegance** in a **typologically diverse** system
  - Can we have one framework for everything? 🌍
  - **Must** a system be elegant?
An analysis of Slavic fronting in HPSG

Open questions:

- Modeling **optional** fronting leads to **spurious** ambiguity
  - How to get rid of it while still accounting for data?
  - **Direction**: information-structure
  - **Gist**: There is **no** optional fronting!
  - Nobody has done this in DELPH-IN HPSG yet

- But isn’t this a problem with **DELPH-IN HPSG** rather than with HPSG or syntactic theory in general?
- **Essential** vs. **incidental** complexity in formalisms
- **Tensions in elegance** in a **typologically diverse** system
- Can we have one framework for everything? 🌍
- **Must** a system be elegant?
- **Part III**: Assembling typologically diverse analyses and evaluating the result
Assembling and evaluating typologically diverse analyses
(4) Gde kto chto vidit?
where who NOM what ACC see 3SG
‘Who sees what where?’[rus]
(4) Gde kto chto vidit?
where who NOM what ACC sees 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\(^1\)
- Input: Typological specification, lexicon, morphological rules
- Output: Implemented HPSG grammar fragment
  - Parse and generate sentences
  - Output syntactic and semantic representations
- Many syntactic phenomena are supported\(^2\)

\(^1\) https://matrix.ling.washington.edu/customize/matrix.cgi
\(^2\) 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

There may be bugs 😓
Matrix library development

- Mapping typological specifications to customized grammar fragments is supported by Matrix libraries 📖
Matrix library development

- Mapping typological specifications to customized grammar fragments is supported by Matrix libraries.
  - Questionnaires are designed based on surveys of typological literature.
Matrix library development

- Mapping typological specifications to customized grammar fragments is supported by Matrix libraries.
  - Questionnaires are designed based on surveys of typological literature.
  - Libraries are evaluated with held-out languages.
Matrix library development

- Mapping typological specifications to customized grammar fragments is supported by Matrix **libraries**
  - Questionnaires are designed based on **surveys of typological literature**
  - 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

- Mapping typological specifications to customized grammar fragments is supported by Matrix libraries
  - Questionnaires are designed based on surveys of typological literature
  - 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
- Latest addition: Constituent questions

(4) Gde kto chto vidit?
where who.NOM what.ACC see.3SG
‘Who sees what where?’ [rus]
**Meta-grammar engineering research question**

**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**
- **Typological scope.**
  - Position of question phrase
    - **Fronting, in situ**
    - Fronting optionality
  - Morphological marking
  - Question particles
    - position
    - obligatoriness
  - Question words

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

Olga Zamaraeva

### Introduction

Computational syntax with HPSG

Assembling typologically diverse analyses

### Future

References

<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>wh-ques-phrase</td>
<td>no</td>
<td>yes</td>
<td>SLASH</td>
</tr>
<tr>
<td></td>
<td>subj-, obj-, adj-ex.</td>
<td>no</td>
<td>no</td>
<td>none</td>
</tr>
<tr>
<td>multi oblig. front.</td>
<td>wh-ques-phrase</td>
<td>no</td>
<td>yes</td>
<td>HDR</td>
</tr>
<tr>
<td></td>
<td>subj-, obj-, adj-ex.</td>
<td>no</td>
<td>no</td>
<td>MODIFIED hasmod</td>
</tr>
<tr>
<td>single opt. front.</td>
<td>wh-ques-phrase</td>
<td>no</td>
<td>yes</td>
<td>SLASH</td>
</tr>
<tr>
<td></td>
<td>in-sutu-phrase</td>
<td>no</td>
<td>yes</td>
<td>none</td>
</tr>
<tr>
<td></td>
<td>subj-, obj-, adj-ex.</td>
<td>no</td>
<td>no</td>
<td>none</td>
</tr>
<tr>
<td>multi opt. front.</td>
<td>wh-ques-phrase</td>
<td>no</td>
<td>yes</td>
<td>MODIFIED hasmod</td>
</tr>
<tr>
<td></td>
<td>in-sutu-phrase</td>
<td>no</td>
<td>yes</td>
<td>HDR</td>
</tr>
<tr>
<td></td>
<td>subj-, obj-, adj-ex.</td>
<td>no</td>
<td>no</td>
<td>HDR</td>
</tr>
<tr>
<td>in situ (no front.)</td>
<td>in-sutu-phrase</td>
<td>no</td>
<td>yes</td>
<td>none</td>
</tr>
</tbody>
</table>

The position of question phrases customization summary

---

25 Zamaraeva 2021
### Assembling particles

<table>
<thead>
<tr>
<th>Specification</th>
<th>Types</th>
<th>Core?</th>
<th>New?</th>
<th>Custom features</th>
</tr>
</thead>
<tbody>
<tr>
<td>clause-final or init.</td>
<td>qpart-comp-lex</td>
<td>no</td>
<td>no</td>
<td>none</td>
</tr>
<tr>
<td></td>
<td>head-comp-phrase</td>
<td>yes</td>
<td>no</td>
<td>INIT</td>
</tr>
<tr>
<td>2nd pos.</td>
<td>ques-clitic-lex</td>
<td>no</td>
<td>yes</td>
<td>none</td>
</tr>
<tr>
<td></td>
<td>non-local</td>
<td>yes</td>
<td>no</td>
<td>YNQ</td>
</tr>
<tr>
<td></td>
<td>basic-binary-phrase</td>
<td>yes</td>
<td>no</td>
<td>L-PERIPH</td>
</tr>
<tr>
<td></td>
<td>int-cl-phrase</td>
<td>no</td>
<td>yes</td>
<td>none</td>
</tr>
<tr>
<td></td>
<td>in-situ-phrase</td>
<td>no</td>
<td>yes</td>
<td>none</td>
</tr>
<tr>
<td>obligatory</td>
<td>in-situ-phrase</td>
<td>no</td>
<td>yes</td>
<td>MC –</td>
</tr>
<tr>
<td>only in embed.</td>
<td>qpart-comp-lex</td>
<td>no</td>
<td>no</td>
<td>MC</td>
</tr>
<tr>
<td>only in polar</td>
<td>qpart-comp-lex</td>
<td>no</td>
<td>no</td>
<td>WH</td>
</tr>
<tr>
<td></td>
<td>ques-clitic-lex</td>
<td>no</td>
<td>yes</td>
<td></td>
</tr>
</tbody>
</table>

Question particles customization summary\

---

Zamaraeva 2021
### Assembling morphological marking

<table>
<thead>
<tr>
<th>Specification</th>
<th>Types</th>
<th>Core?</th>
<th>New?</th>
<th>Custom features</th>
</tr>
</thead>
<tbody>
<tr>
<td>Same mkg for polar and wh-</td>
<td>interrog-lex-rule</td>
<td>no</td>
<td>yes</td>
<td>none</td>
</tr>
<tr>
<td></td>
<td>indicative-lex-rule</td>
<td>no</td>
<td>yes</td>
<td>none</td>
</tr>
<tr>
<td>Separate paradigms</td>
<td>indicative-lex-rule</td>
<td>no</td>
<td>yes</td>
<td>none</td>
</tr>
<tr>
<td></td>
<td>polar-lex-rule</td>
<td>no</td>
<td>yes</td>
<td>none</td>
</tr>
<tr>
<td></td>
<td>wh-subj-lex-rule</td>
<td>no</td>
<td>yes</td>
<td>none</td>
</tr>
<tr>
<td></td>
<td>wh-obj-lex-rule</td>
<td>no</td>
<td>yes</td>
<td>none</td>
</tr>
</tbody>
</table>

Morphological question marking 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

<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><em>in situ</em>, particle, int. verbs</td>
</tr>
</tbody>
</table>

- Single fronting, particles, morphological marking, interrogative verbs
- Question words as predicates
- Did not come up: Multiple fronting and LDDs
Evaluating the constituent questions library

<table>
<thead>
<tr>
<th>Language</th>
<th>ISO-639-3</th>
<th>Family</th>
<th>Gram. items</th>
<th>Coverage%</th>
<th>Handled phenomena</th>
</tr>
</thead>
<tbody>
<tr>
<td>Apinajé</td>
<td>[apn]</td>
<td>Macro-Jê</td>
<td>17</td>
<td>82.3</td>
<td>evidentials, arg. drop, clausal modifiers</td>
</tr>
<tr>
<td>Makah</td>
<td>[myh]</td>
<td>Wakashan</td>
<td>14</td>
<td>78.5</td>
<td>clausal complements, arg. drop</td>
</tr>
<tr>
<td>Pacoh</td>
<td>[pac]</td>
<td>Austroasiatic</td>
<td>26</td>
<td>84.6</td>
<td>arg. drop</td>
</tr>
<tr>
<td>Paresi-Haliti</td>
<td>[pab]</td>
<td>Arawakan</td>
<td>64</td>
<td>56.0</td>
<td>focus, adnom. poss, coordination</td>
</tr>
<tr>
<td>Jalkunan</td>
<td>[bxl]</td>
<td>Mande</td>
<td>33</td>
<td>78.8</td>
<td>adnom. poss.</td>
</tr>
</tbody>
</table>

✓ Single fronting, particles, morphological marking, interrogative verbs

✗ Question words as predicates

– Did not come up: Multiple fronting and LDDs

柬埔：in interaction with **information structure** and clausal complements

Zamaraeva 2021
Interim Summary

- The Grammar Matrix facilitates grammar creation for a wide variety of languages.
Interim Summary

- The Grammar Matrix facilitates grammar creation for a wide variety of languages.

- Bigger grammar fragments now possible to obtain from the Matrix\(^{30}\)
  - Further projects focused on polysynthetic languages are needed
  - What’s next?
Future directions
Applying grammars in Natural Language Processing

- Ultimately, NLP relies on formal annotation
  - ERG is more robust than PTB/PSD and can be automatically rerun
  - We need this for more languages as a long-term investment
Applying grammars in Natural Language Processing

- Ultimately, NLP relies on formal annotation
  - ERG is more robust than PTB/PSD and can be automatically rerun
  - We need this for more languages as a long-term investment
  - Meanwhile...
Applying grammars in Natural Language Processing

- Ultimately, NLP relies on formal annotation
  - ERG is more robust than PTB/PSD and can be automatically rerun
  - We need this for more languages as a long-term investment
  - Meanwhile...

- NLP for grammar coaching

- Statistical systems are imprecise 👏
Applying grammars in Natural Language Processing

- Ultimately, NLP relies on formal annotation
  - ERG is more robust than PTB/PSD and can be automatically rerun
  - We need this for more languages as a long-term investment
  - Meanwhile...

- NLP for grammar coaching

- Statistical systems are imprecise 👏
- Adding more training data still won’t help with explanations
Applying grammars in Natural Language Processing

- Ultimately, NLP relies on formal annotation
  - ERG is more robust than PTB/PSD and can be automatically rerun
  - We need this for more languages as a long-term investment
  - Meanwhile...

- NLP for grammar coaching
  - Statistical systems are imprecise
  - Adding more training data still won’t help with explanations
  - Grammars: Incorporate L2 productions

---

Schneider and McCoy 1998
Assembling Syntax

Olga Zamaraeva

Introduction

Computational syntax with HPSG

Assembling typologically diverse analyses

Future

References

Applying grammars in Natural Language Processing

- Ultimately, NLP relies on formal annotation
  - ERG is more robust than PTB/PSD and can be automatically rerun
  - We need this for more languages as a long-term investment
  - Meanwhile...

- NLP for grammar coaching

  ![The dog like the cat]

  Add a hyphen

  **dog-like**

- Statistical systems are imprecise

- Adding more training data still won’t help with explanations

- Grammars: Incorporate L2 productions

---

31 Schneider and McCoy 1998

Ultimately, NLP relies on formal annotation

ERG is more robust than PTB/PSD and can be automatically rerun

We need this for more languages as a long-term investment

Meanwhile...

NLP for grammar coaching

Statistical systems are imprecise

Adding more training data still won’t help with explanations

Grammars: Incorporate L2 productions

---

31 Schneider and McCoy 1998
Applying grammars in Natural Language Processing

- Ultimately, NLP relies on formal annotation
  - ERG is more robust than PTB/PSD and can be automatically rerun
  - We need this for more languages as a long-term investment
  - Meanwhile...

- NLP for grammar coaching
  - The dog like the cat
    - Add a hyphen
    - dog-like

- Statistical systems are imprecise
- Adding more training data still won’t help with explanations
- Grammars: Incorporate L2 productions
- Map L2 productions to useful feedback

---

31 Schneider and McCoy 1998
HPSG grammars for grammar coaching

- First step: Spanish and Galician
  - with Gómez Rodríguez and Alonso Ramos, U. of A Coruña
- Next step: More languages
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 🦜
Assembling Syntax
Olga Zamaraeva

Introduction
Computational syntax with HPSG
Assembling typologically diverse analyses
Future

References


References

References


Bender, Emily M, Dan Flickinger, Stephan Oepen, et al. (2004). “Arboretum: Using a precision grammar for grammar checking in CALL”. In: InSTIL/ICALL.
