Multiple Question Fronting without Relational Constraints An Analysis of Russian as a Basis for Cross-Linguistic Modeling

27th International Conference on Head-Driven Phrase Structure Grammar

Olga Zamaraeva, Guy Emerson
University of Washington
University of Cambridge

August 17 2020

Introduction

- In Slavic languages, multiple question phrases can be fronted
- We present an implementation of multiple extraction in DELPH-IN HPSG
  - Unification is the only natively defined operation
  - No relational constraints (e.g. native list append)
- We test the implementation on a test suite of Russian sentences
- The implementation served as basis for the Constituent Questions library for the Grammar Matrix customization system

Russian question fronting

(1) *Kogda kto kogo videl?*
when who.NOM who.ACC see.PST

'When did which person saw which other person?' [rus]

(2) *Kogda kto kogo ty tochno znaesh (chto) videl?*
when who.NOM who.ACC 2SG for.sure know (that) see.PST

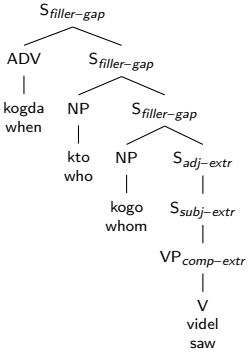
'When do you know for sure who saw whom?' ('What are the sets of times and persons such that one person saw another at a certain time, such that you know this set of facts for sure?') [rus]

Fronting multiple phrases in DELPH-IN

- The order of the daughters is fixed in the phrase structure rule
- Filler-gap rule needs to apply multiple times
- Extraction rules need to apply to each other

(1) *Kogda kto kogo videl?*
when who.NOM who.ACC see.PST

'When did which person saw which other person?' [rus]



Fronting multiple phrases in DELPH-IN

- In most previous DELPH-IN work,[1] nonlocal features SLASH and QUE were restricted to *difference lists* of length 1
  - Multiple extraction was unnecessary for e.g. the English Resource Grammar
  - Indefinite recursion of adjunct extraction avoided

$$\begin{bmatrix} \textit{filler-gap-phrase} & \\ \text{SLASH} & \langle !\ ! \rangle \\ \text{ARGS} & \left\langle \boxed{2}, \begin{bmatrix} \text{SLASH} \langle !\ \boxed{2}\ ! \rangle \end{bmatrix} \right\rangle \end{bmatrix}$$

[1] Flickinger 2000; Bender, Flickinger, and Oepen 2002, *inter alia*.

Fronting multiple phrases in DELPH-IN

- Our analysis:
  - Allow recursive application of the *filler-gap-phrase*[2]

$$\begin{bmatrix} \textit{filler-gap-phrase} & \\ \text{SLASH} & \boxed{1} \\ \text{ARGS} & \left\langle \boxed{2}, \left[\text{SLASH} \middle| \text{LIST} \left\langle \boxed{2} \right\rangle \oplus \boxed{1} \right] \right\rangle \end{bmatrix}$$

- How does $\oplus$ work for SLASH?
  - Possible to do with *diff-list*-valued SLASH[3] but cumbersome
  - We replace *difference list* with a new type *append list*[4]

[2] Sag, Wasow, and Bender 2003; Crysmann 2015.

[3] Flickinger 2000; Crysmann 2015.

[4] Emerson 2017; Emerson 2019; Aguila-Multner and Crysmann 2018.

Append-lists

- We want to:
  1. Append lists to each other
  2. Use closed lists (e.g. to check length)

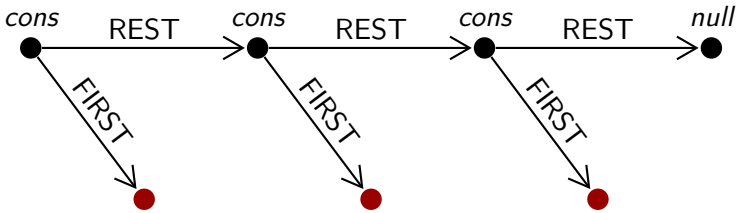
Append-lists

- We want to:
  1. Append lists to each other
  2. Use closed lists (e.g. to check length)
- Like diff-lists, append-lists contain lists
- Unlike diff-lists, these lists are closed

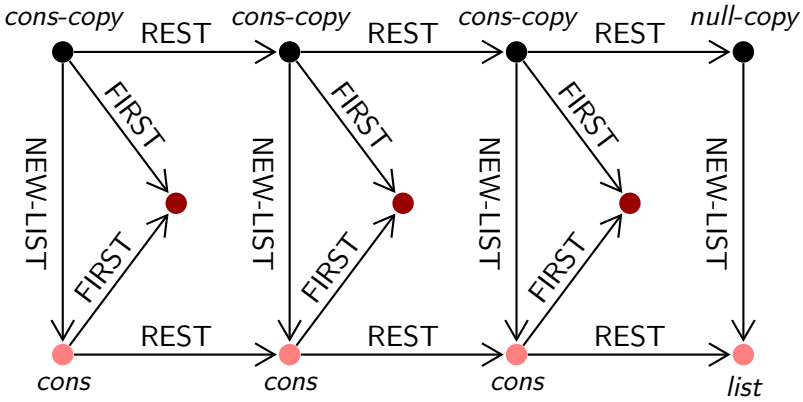
Append-lists

- We want to:
  1. Append lists to each other
  2. Use closed lists (e.g. to check length)
- Like diff-lists, append-lists contain lists
- Unlike diff-lists, these lists are closed
- The append operation:
  1. Creates an open list from each closed list
  2. Combines the open lists (like diff-lists)
  3. Closes the list

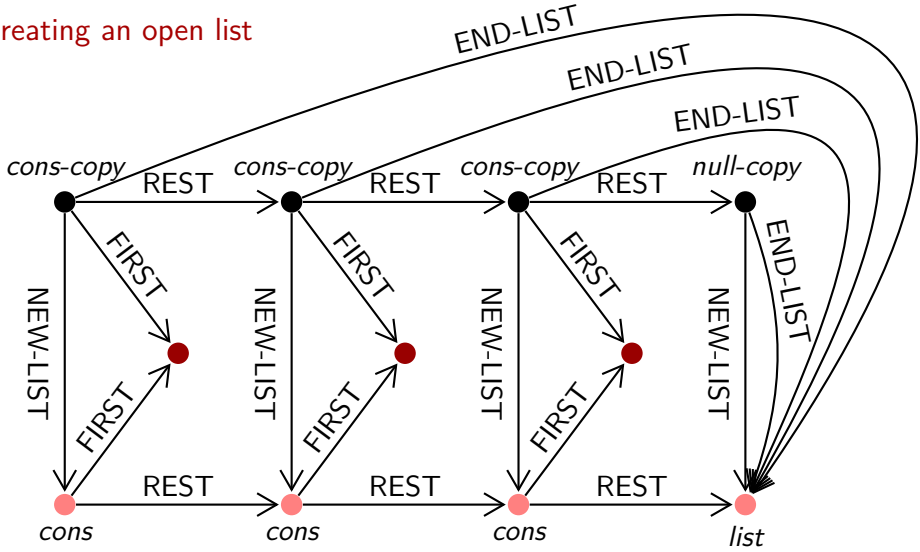
Creating an open list



Creating an open list



Creating an open list



Append-lists in a grammar

- A type to use instead of *diff-list*
- Feature APPEND allows for easy append
- Feature PLACEHOLDER separates specifying an element and appending it (useful for adding adjuncts to the SLASH list)

$$\begin{bmatrix} \textit{append-list} & \\ \text{LIST} & \boxed{0}\textit{list} \\ \text{APPEND} & \begin{bmatrix} \textit{list-of-append-lists} \\ \text{APPEND-RESULT } \boxed{0} \end{bmatrix} \\ \text{PLACEHOLDER}|\text{LIST} & \langle [] \rangle \end{bmatrix}$$

Fronting multiple questions with append-lists

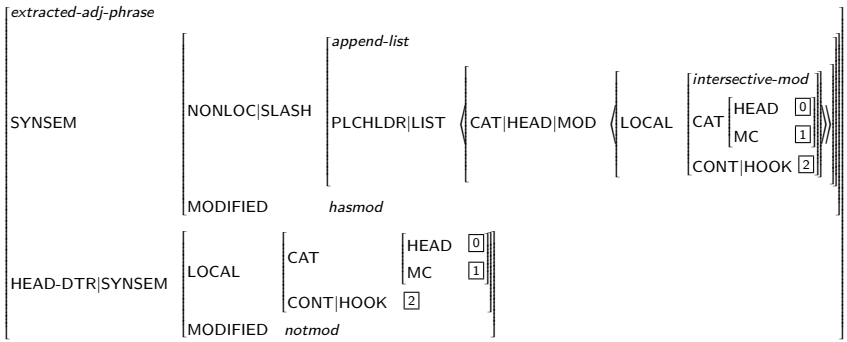
- Our analysis:
  - Recursive *filler-gap* rule[5]

$$\left[\begin{array}{l} \textit{wh-ques-phrase} \\ \text{SYNSEM} \left[\begin{array}{ll} \text{LOCAL|CAT} & \left[\text{HEAD} \quad \textit{verb}\right] \\ \text{NON-LOCAL} & \left[\begin{array}{ll} \text{SLASH} & \boxed{1} \\ \text{QUE|LIST} & \langle\rangle \end{array}\right] \end{array}\right] \\ \text{HEAD-DTR|SYNSEM} \left[\text{NON-LOCAL|SLASH|LIST} \quad \left\langle \boxed{2} \right\rangle \oplus \boxed{1}\right] \\ \text{NON-HEAD-DTR|SYNSEM} \left[\begin{array}{ll} \text{LOCAL} & \boxed{2} \\ \text{NON-LOCAL|QUE|LIST} & \langle \textit{ref-ind} \rangle \end{array}\right] \end{array}\right]$$

[5] Sag, Wasow, and Bender 2003; Crysmann 2015.

Fronting multiple questions with append-lists

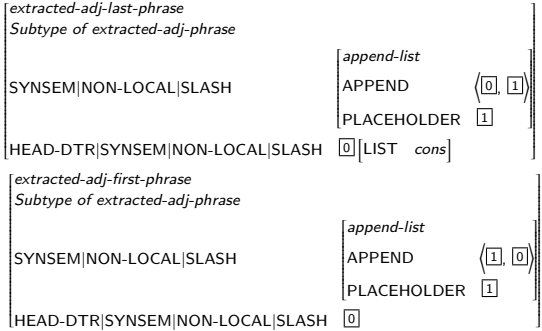
- Our analysis:
  - Use MODIFIED feature to curb infinite recursion of adjunct extraction[6]
  - The PLACEHOLDER feature allows for an adjunct to be prepended or appended to the SLASH list



6 Flickinger 2000.

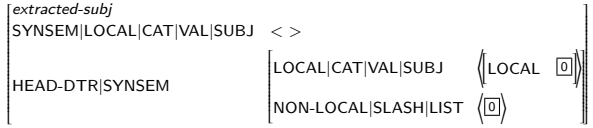
Fronting multiple questions with append-lists

- Our analysis:
  - The PLACEHOLDER feature allows for an adjunct to be prepended or appended to the SLASH list



Fronting multiple questions with append-lists

- Argument extraction remains largely as implemented in the English Resource Grammar[7] and in the Grammar Matrix[8]
- "Lexical threading"[9]
  - Verbs "amalgamate" their arguments' SLASH and QUE



7 Flickinger 2000.

8 Bender, Flickinger, and Oepen 2002.

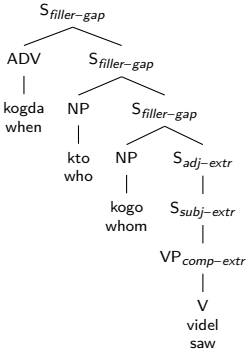
9 Bouma, Malouf, and Sag 2001.

Fronting multiple questions with append-lists

$$\begin{bmatrix} \textit{wh-phrase} \\ \text{SLASH} & \boxed{1} \\ \text{ARGS} & \left\langle \boxed{2}, \left[\text{SLASH} \left\langle \boxed{2} \right\rangle \oplus \boxed{1} \right] \right\rangle \end{bmatrix}$$

(1) *Kogda kto kogo videl?*
when who.NOM who.ACC see.PST

'When did which person saw which other person?' [rus]



Constituent questions in the Grammar Matrix

- We integrated our analysis in the Grammar Matrix system[10]
  - Given a typological and lexical specification, output a grammar automatically
- Tested so far on Russian, English, Japanese, Yukaghir, Chukchi specifications
  - Russian and English use fronting

[10] Bender, Flickinger, and Oepen 2002; Bender, Drellishak, et al. 2010.

Testing on Russian

Coverage %	Overgeneration %	Ambiguity
79.8	5.8	2.13

- 284 sentences; declarative and interrogative
- Constructed to represent a variety of possible (198) and impossible (86) syntactic patterns related to questions
- Minimal lexicon

(3)

V	*kakoi*	*Ivan*	*priehal*	*gorod?*
IN	which.SG.ACC	Ivan.NOM	arrive.PAST.3SG	town.SG.ACC

'In which town did Ivan arrive?" [rus]

(4)

??Kogda	*gde*	*my*	*kupili*	*eti*	*knigi?*
when	where	1PL.NOM	buy.PAST.1PL	this.PL.ACC	book.PL.ACC

Intended: 'When [and] where did we buy these books?'

Does Russian have optional fronting?

(5) *Ty gde rabotaesh?*
2SG where work.2SG

'Where do you work?' [rus]

- Can declarative phrasal rules take QUE-nonemtpy daughters?

(4) *??Kogda gde my kupili eti knigi?*
when where 1PL.NOM buy.PAST.1PL this.PL.ACC book.PL.ACC

Intended: 'When [and] where did we buy these books?'

Conclusion

- Multiple extraction can be modeled straightforwardly in DELPH-IN formalism with the use of the new *append-list* type
- The analysis served as basis for cross-linguistic modeling of fronting languages in the Grammar Matrix
- The next step is to model various word orders
  - Russian motivates modeling flexible orders of question phrases via obligatory extraction (rather than allowing declarative phrasal types to take QUE-nonempty daughters)
  - In cases where the question phrase appears *in situ*, topicalization fronting may be occurring

References

Aguila-Multner, Gabriel and Berthold Crysmann (2018). "Feature Resolution by Lists: The Case of French Coordination". In: *Formal Grammar. 23rd International Conference*. Ed. by Annie Foret, Greg Kobele, and Sylvain Pogodalla. Lecture Notes in Computer Science. Heidelberg: Springer, pp. 1–15.

Bender, Emily M, Scott Drellishak, et al. (2010). "Grammar Customization". In: *Research on Language & Computation* 8.1. 10.1007/s11168-010-9070-1, pp. 23–72. ISSN: 1570-7075. URL: http://dx.doi.org/10.1007/s11168-010-9070-1.

Bender, Emily M, 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: *Proceedings of the Workshop on Grammar Engineering and Evaluation at the 19th International Conference on Computational Linguistics*. Ed. by John Carroll, Nelleke Oostdijk, and Richard Sutcliffe. Taipei, Taiwan, pp. 8–14.

Bouma, Gosse, Robert Malouf, and Ivan A Sag (2001). "Satisfying Constraints on Extraction and Adjunction". In: *Natural Language & Linguistic Theory* 19.1, pp. 1–65.

Crysmann, Berthold (2015). "Resumption and Extraction in an Implemented HPSG grammar of Hausa". In: *Proceedings of the Grammar Engineering Across Frameworks (GEAF) 2015 Workshop*, pp. 65–72.

Emerson, Guy (2017). "List Appends in TDL". Delph-in Summit.

— (2019). "Wrapper types: relational constraints without relational constraints". Delph-in Summit.

Flickinger, Dan (2000). "On building a more effcient grammar by exploiting types". In: *Natural Language Engineering* 6.01, pp. 15–28.

Sag, Ivan A, Thomas Wasow, and Emily M Bender (2003). *Syntactic theory: A formal introduction*. Center for the Study of Language and Information Stanford, CA.