Exact and Efficient Graph Parsing

Weiwei Sun

Institute of Computer Science
and Technology
Peking University

July 16, 2019

Joint work with

Yufei is with you

Fashion

A growing interest in semantic representations

- Bi-lexical Semantic Dependency Graphs
- Abstract Meaning Representations
- Elementary Dependency Structures
- Dependency-based Minimal Recursion Semantics
- Universal Conceptual Cognitive Annotation

Fashion

A growing interest in semantic representations

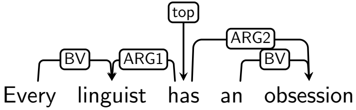
- Bi-lexical Semantic Dependency Graphs
- Abstract Meaning Representations
- Elementary Dependency Structures
- Dependency-based Minimal Recursion Semantics
- Universal Conceptual Cognitive Annotation

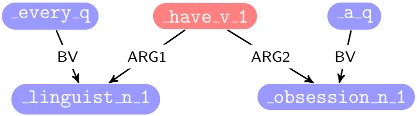
Fashion

A growing interest in semantic representations

- Bi-lexical Semantic Dependency Graphs
- Abstract Meaning Representations
- Elementary Dependency Structures
- Dependency-based Minimal Recursion Semantics
- Universal Conceptual Cognitive Annotation

Many descriptive and theoretical differences, but one important similarity: All use graphs!





New questions

> *Work in computational linguistics is in some cases motivated from a scientific perspective in that one is trying to provide a computational explanation for a particular linguistic or psycholinguistic phenomenon; and in other cases the motivation may be more purely technological in that one wants to provide a working component of a speech or natural language system.*
>
> *www.aclweb.org*

New questions

> *Work in computational linguistics is in some cases motivated from a scientific perspective in that one is trying to provide a computational explanation for a particular linguistic or psycholinguistic phenomenon; and in other cases the motivation may be more purely technological in that one wants to provide a working component of a speech or natural language system.*
>
> *www.aclweb.org*

1. How can we build a high-performance string-to-graph parser?
2. How can we build a high-performance graph-to-string parser?

New questions

> *Work in computational linguistics is in some cases motivated from a scientific perspective in that one is trying to provide a computational explanation for a particular linguistic or psycholinguistic phenomenon; and in other cases the motivation may be more purely technological in that one wants to provide a working component of a speech or natural language system.*
>
> *www.aclweb.org*

1. How can we build a high-performance string-to-graph parser?
2. How can we build a high-performance graph-to-string parser?

3. Can we use a single model to achieve the two goals?
4. Is our model linguistically meaningful?
5. Can we apply our model to evaluate a linguistic hypothesis?

Outline

Graph-Based Meaning Representation

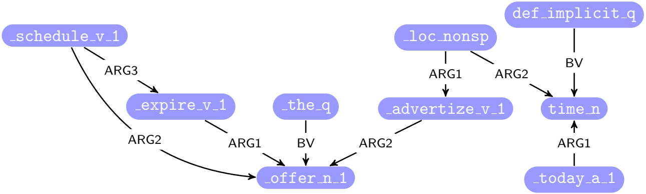
Synchronous Hyperedge Replacement Grammar

Parsing a Graph with an SHRG

Comparative Computational Semantics

Reflections on predicate–argument structure

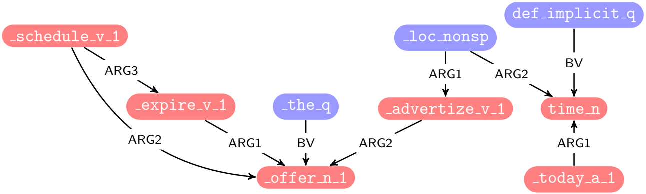
- Arguments recursively are predicates most of the time;



The offer advertized today is scheduled to expire. [WSJ #0032002]

Reflections on predicate–argument structure

- Arguments recursively are predicates most of the time;
- ‘content words’ introduce predicates, e.g. nouns, verbs, and adjs;

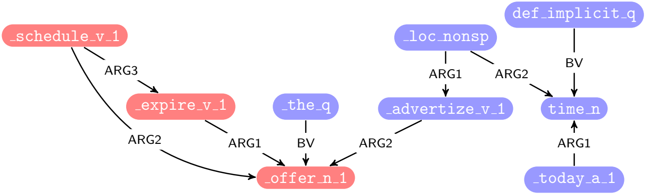


The offer advertized today is scheduled to expire. [WSJ #0032002]

Reflections on predicate–argument structure

- Arguments recursively are predicates most of the time;
- ‘content words’ introduce predicates, e.g. nouns, verbs, and adjs;
- arity determines the (possible) number of arguments;
- seemingly zero-place predicates can have referential argument;

$$\text{offer'}(x) \wedge \text{schedule'}(\_, x, \text{expire'}(\_, x))$$



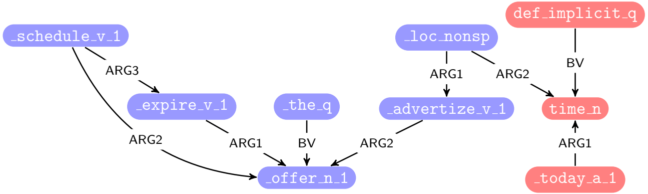
The offer advertized today is scheduled to expire. [WSJ #0032002]

Reflections on predicate–argument structure

- Arguments recursively are predicates most of the time;
- ‘content words’ introduce predicates, e.g. nouns, verbs, and adjs;
- arity determines the (possible) number of arguments;
- seemingly zero-place predicates can have referential argument;

$$\text{offer'}(x) \wedge \text{schedule'}(\_, x, \text{expire'}(\_, x))$$

- possibly multiple predicates per word or construction.



The offer advertized today is scheduled to expire. [WSJ #0032002]

Why graphs

Allow us to exploit graph-centric

- visualization,
- formalisms,
- algorithms,
- neural architectures
- and many other things

to build an accurate mapping between natural language utterances and in-depth meaning representations.

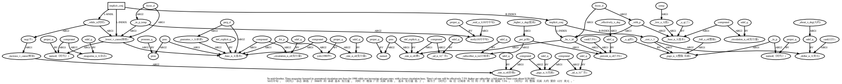
Two fundamental problems

In mid-October, Time magazine lowered its guaranteed circulation rate base for 1990 while not increasing ad page rates; with a lower circulation base, Time's ad rate will be effectively 7.5% higher per subscriber; a full page in Time costs about \$120,000.

Two fundamental problems

In mid-October, Time magazine lowered its guaranteed circulation rate base for 1990 while not increasing ad page rates; with a lower circulation base, Time's ad rate will be effectively 7.5% higher per subscriber; a full page in Time costs about $120,000.

Natural Language Understanding ⇒ String-to-graph Parsing



Neural string-to-graph parsers are cool!

Elementary Dependency Structure	SMATCH	EDM
Factorization-Based	95+	-
Synchronous Hyperedge Replacement Grammar	93+	92+

Neural string-to-graph parsers are cool!

Elementary Dependency Structure	SMATCH	EDM
Factorization-Based	95+	-
Synchronous Hyperedge Replacement Grammar	93+	92+

Do they touch the upper bound?

	Annotator Comparison			
Metric	A vs. B	A vs. C	B vs. C	Average
EDM	94	94	95	94

E. Bender, D. Flickinger, S. Oepen, W. Packard and A. Copestake. 2015. Layers of Interpretation: On Grammar and Compositionality.

Neural string-to-graph parsers are cool!

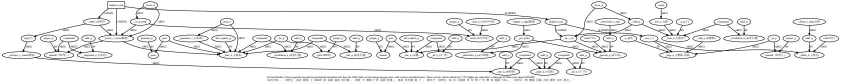
Elementary Dependency Structure	SMATCH	EDM
Factorization-Based	95+	-
Synchronous Hyperedge Replacement Grammar	93+	92+

Do they touch the upper bound?

	Annotator Comparison			
Metric	A vs. B	A vs. C	B vs. C	Average
EDM	94	94	95	94

E. Bender, D. Flickinger, S. Oepen, W. Packard and A. Copestake. 2015. Layers of Interpretation: On Grammar and Compositionality.

Two fundamental problems



Natural Language Generation ⇒ Graph-to-string Parsing

10月中旬,《时代》杂志降低了1990年的承诺基本发行量,同时不增加广告页面价格。基本发行量低了,相当于《时代》每位订阅者所付的广告费将提高7.5%。

Beyond building practical systems

> *Work in computational linguistics is in some cases motivated from a scientific perspective in that one is trying to provide a computational explanation for a particular linguistic or psycholinguistic phenomenon; and in other cases the motivation may be more purely technological in that one wants to provide a working component of a speech or natural language system.*
>
> *www.aclweb.org*

Beyond building practical systems

Work in computational linguistics is in some cases motivated from a scientific perspective in that one is trying to provide a computational explanation for a particular linguistic or psycholinguistic phenomenon; and in other cases the motivation may be more purely technological in that one wants to provide a working component of a speech or natural language system.

www.aclweb.org

1 How can we build a high-performance string-to-graph parser?

⇒ How can we build a high-performance graph-to-string parser?

Beyond building practical systems

Work in computational linguistics is in some cases motivated from a scientific perspective in that one is trying to provide a computational explanation for a particular linguistic or psycholinguistic phenomenon; and in other cases the motivation may be more purely technological in that one wants to provide a working component of a speech or natural language system.

www.aclweb.org

1 How can we build a high-performance string-to-graph parser?

⇒ How can we build a high-performance graph-to-string parser?

3 Can we use a single model to achieve the two goals?

4 Is our model linguistically meaningful?

⇒ Can we apply our model to evaluate a linguistic hypothesis?

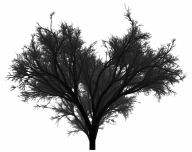
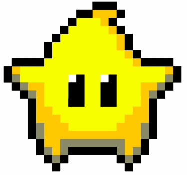
Outline

Context-free rewriting

Context-free rewriting is a powerful way to build complex things

Context-free rewriting

Context-free rewriting is a powerful way to build complex things

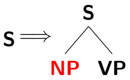
https://www.contextfreeart.org/

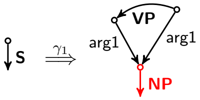
Syntactico-semantic composition as CF rewriting

S

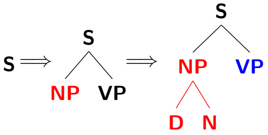
↓ S

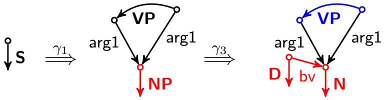
Syntactico-semantic composition as CF rewriting



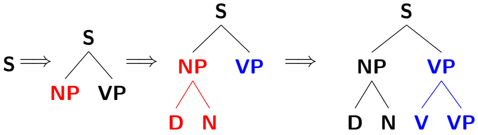


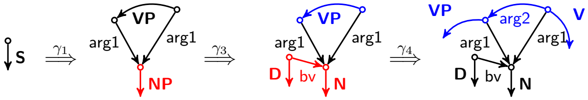
Syntactico-semantic composition as CF rewriting



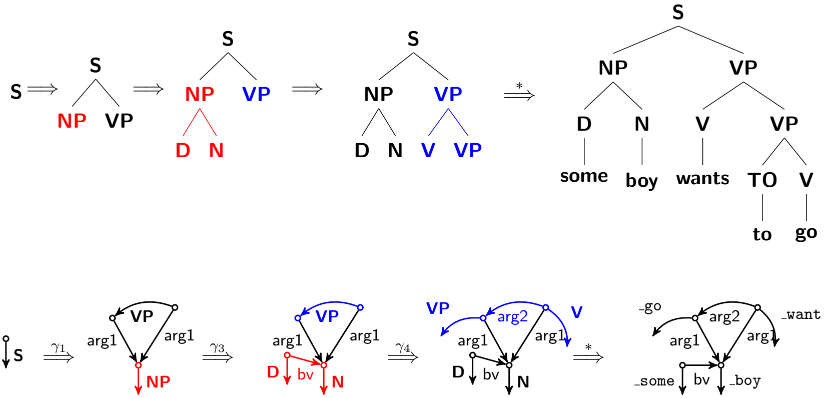


Syntactico-semantic composition as CF rewriting

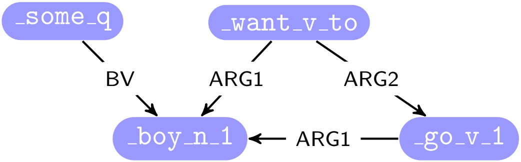




Syntactico-semantic composition as CF rewriting



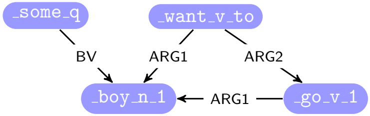
Hypergraph

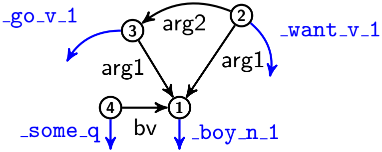


A graph consists of:

- A set of nodes.
- A set of edges connecting two nodes.

Hypergraph





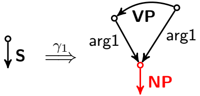
A hypergraph adds:

- Hyperedges connecting any number of nodes.
- A single node can be treated as an edge.

Hyperedge Replacement Grammar

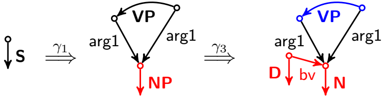
- Terminal vs. non-terminal hyperedges
- Non-terminal hyperedges are utilized to control a derivation process.
- A derivation starts from a non-terminal hyperedge .

Hyperedge Replacement Grammar



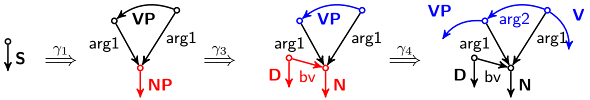
- Terminal vs. non-terminal hyperedges
- Non-terminal hyperedges are utilized to control a derivation process.
- A derivation starts from a non-terminal hyperedge .
- In a derivation step, we substitute a non-terminal hyperedge with a hypergraph .

Hyperedge Replacement Grammar



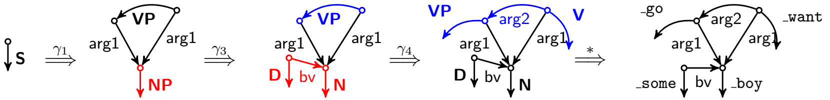
- Terminal vs. non-terminal hyperedges
- Non-terminal hyperedges are utilized to control a derivation process.
- A derivation starts from a non-terminal hyperedge .
- In a derivation step, we substitute a non-terminal hyperedge with a hypergraph .

Hyperedge Replacement Grammar



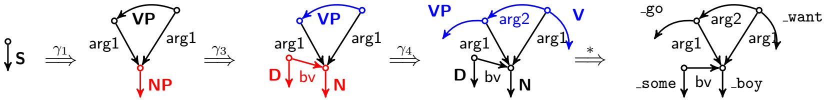
- Terminal vs. non-terminal hyperedges
- Non-terminal hyperedges are utilized to control a derivation process.
- A derivation starts from a non-terminal hyperedge .
- In a derivation step, we substitute a non-terminal hyperedge with a hypergraph .

Hyperedge Replacement Grammar



- Terminal vs. non-terminal hyperedges
- Non-terminal hyperedges are utilized to control a derivation process.
- A derivation starts from a non-terminal hyperedge .
- In a derivation step, we substitute a non-terminal hyperedge with a hypergraph .
- We repeat until all edges are terminal ones.

Hyperedge Replacement Grammar

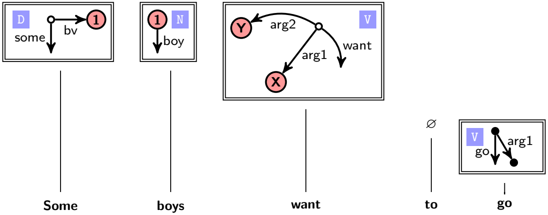


- Terminal vs. non-terminal hyperedges (symbols)
- Non-terminal hyperedges (symbols) are utilized to control a derivation process.
- A derivation starts from a non-terminal hyperedge (symbol).
- In a derivation step, we substitute a non-terminal hyperedge (symbols) with a hypergraph (a sequence of symbols).
- We repeat until all edges (symbols) are terminal ones.

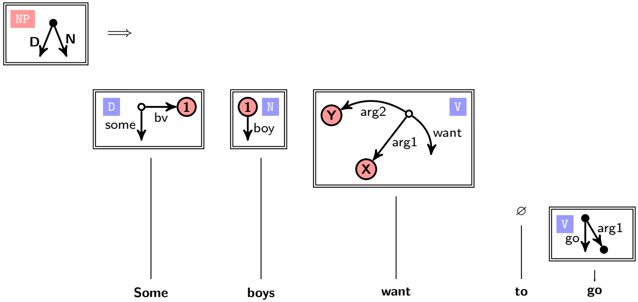
As a bottom-up graph gluing procedure

Some **boys** **want** **to** **go**

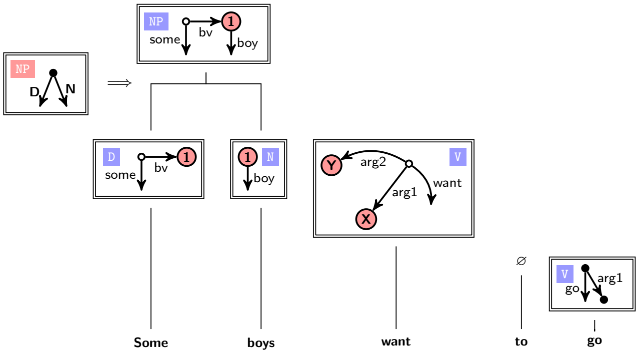
As a bottom-up graph gluing procedure



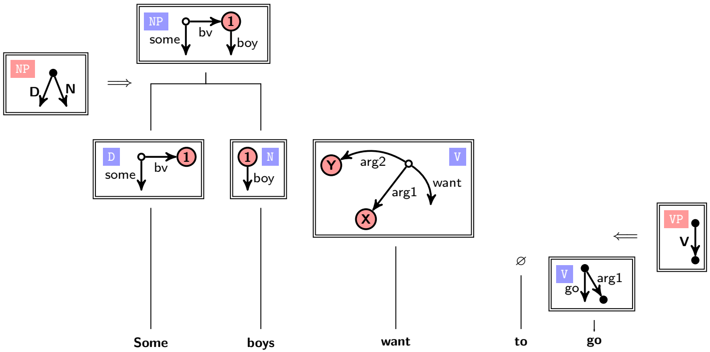
As a bottom-up graph gluing procedure



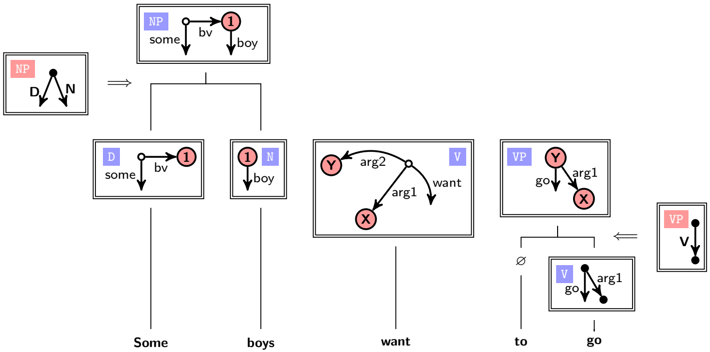
As a bottom-up graph gluing procedure



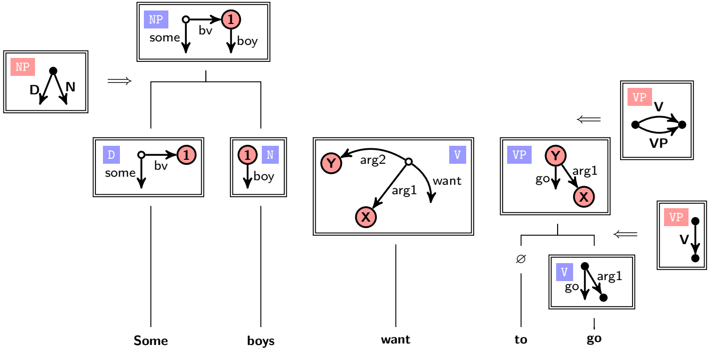
As a bottom-up graph gluing procedure



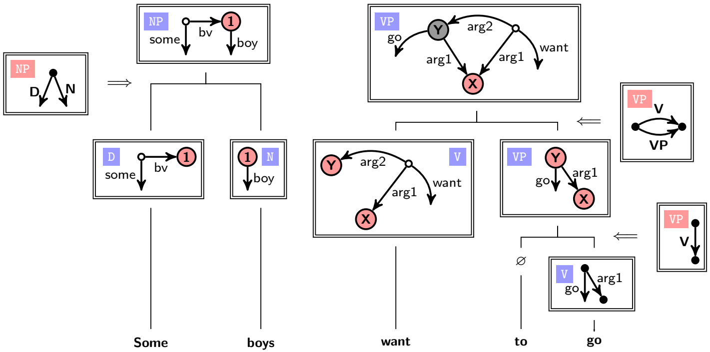
As a bottom-up graph gluing procedure



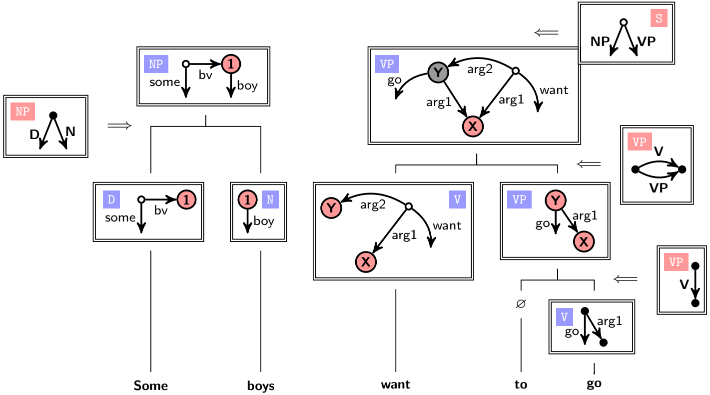
As a bottom-up graph gluing procedure



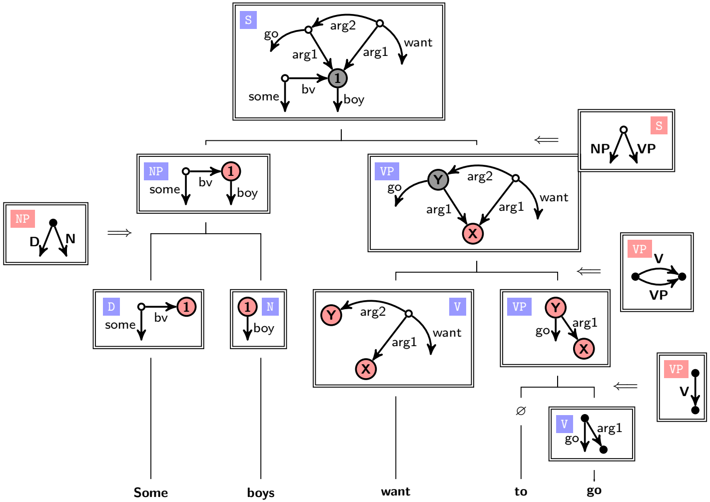
As a bottom-up graph gluing procedure



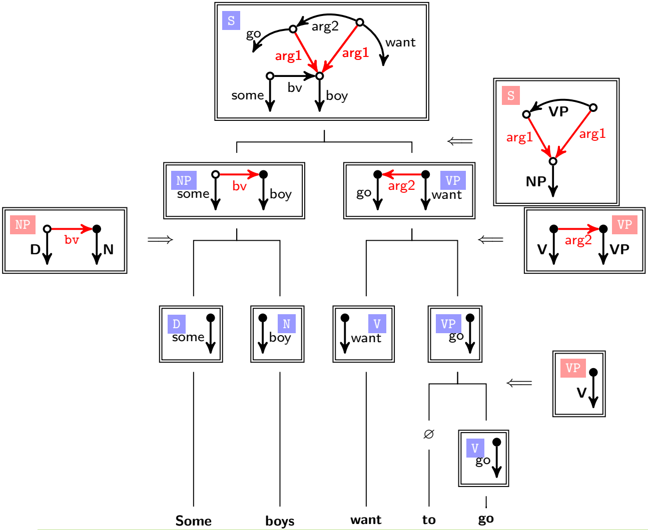
As a bottom-up graph gluing procedure



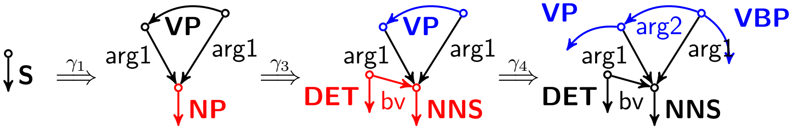
As a bottom-up graph gluing procedure



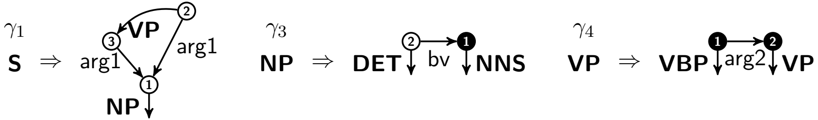
Flexibility



HRGs can be linguistically meaningful



Construction semantics



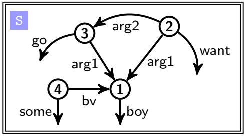
Control construction (Equi verbs)

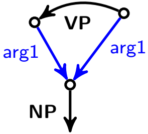
Generalized quantifier

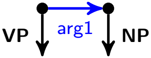
Predicate–argument structure

Outline

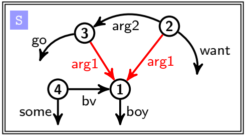
A top-down strategy



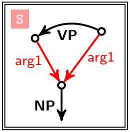


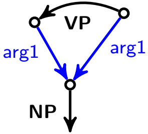


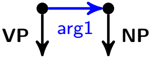
A top-down strategy



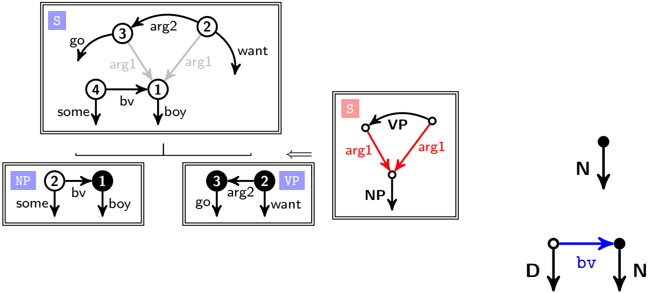
⟸



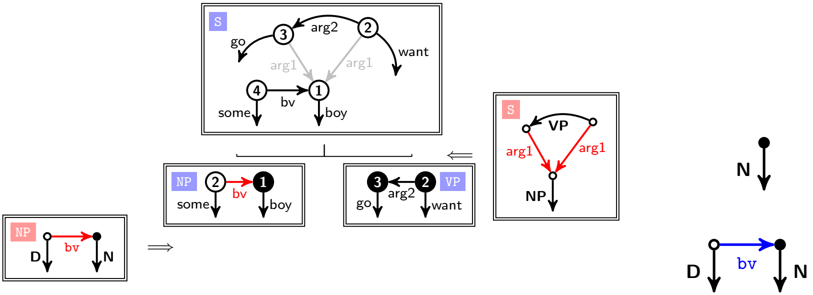




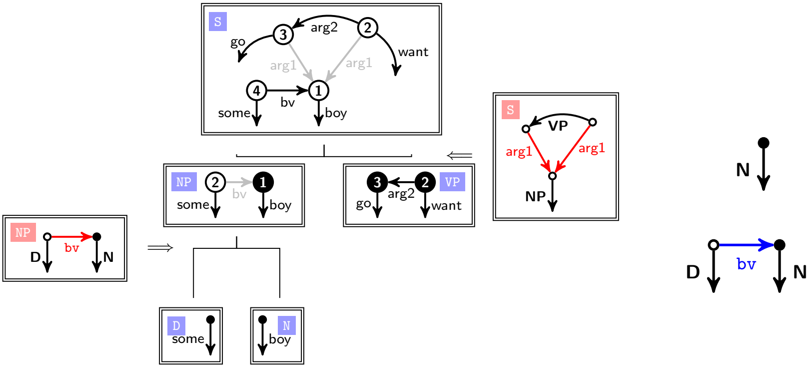
A top-down strategy



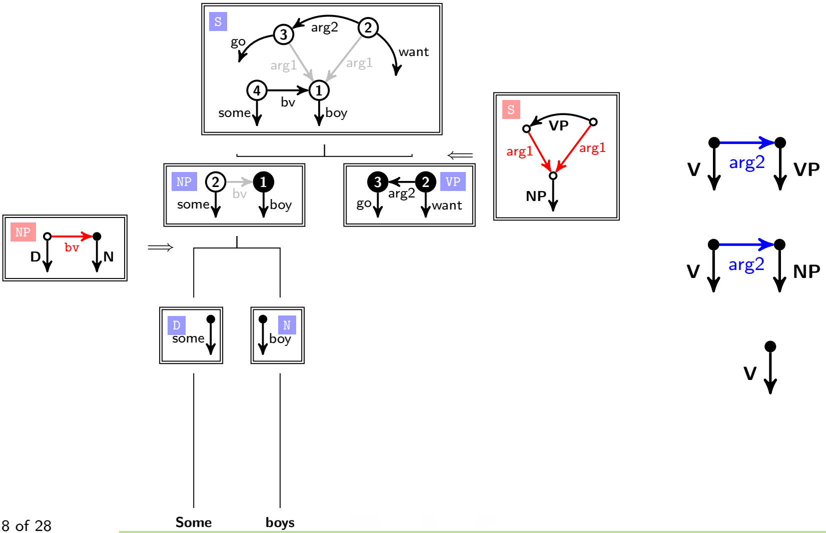
A top-down strategy



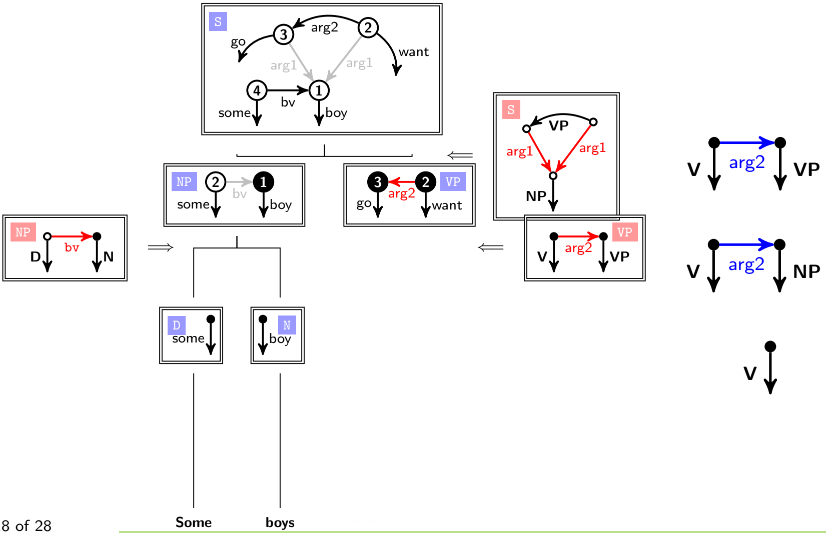
A top-down strategy



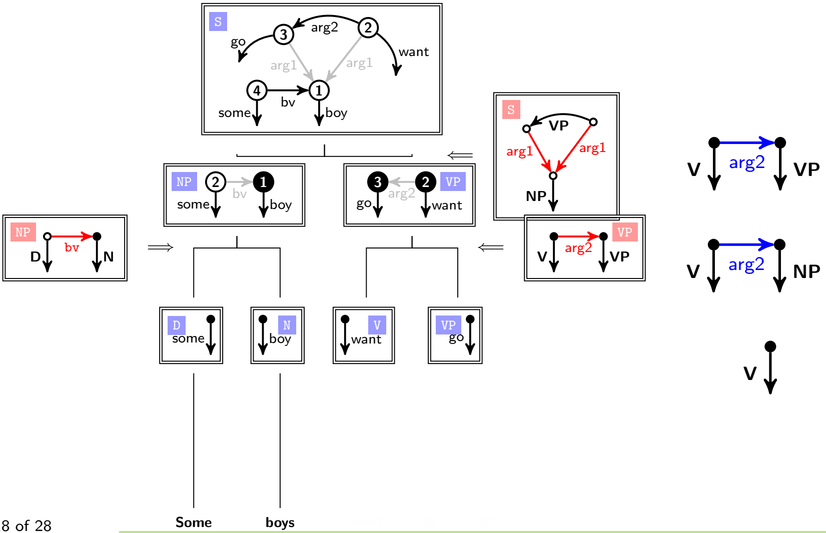
A top-down strategy



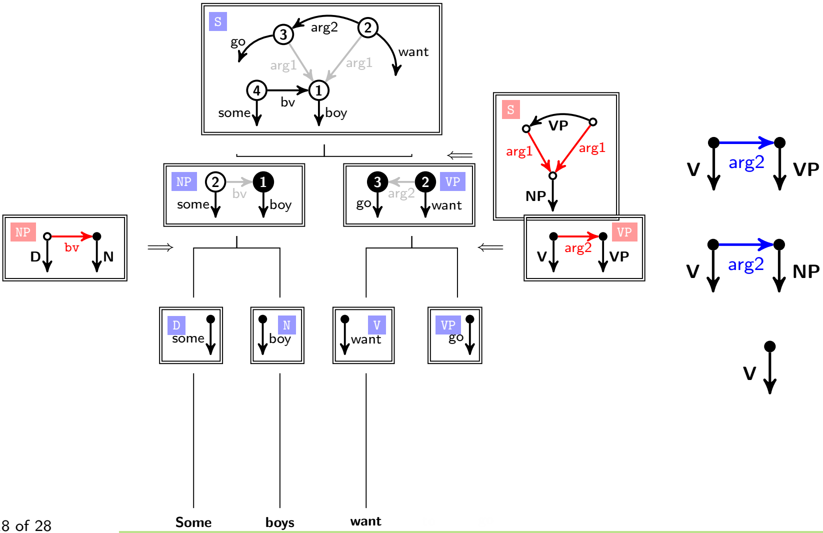
A top-down strategy



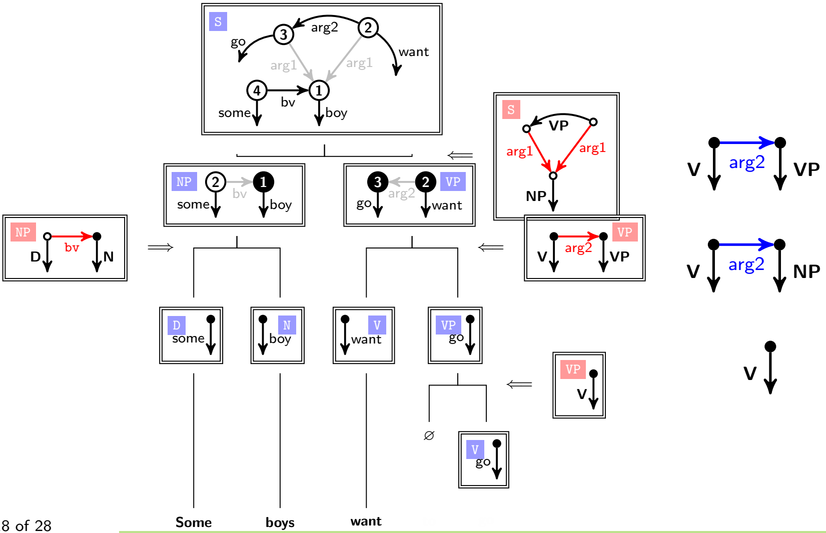
A top-down strategy



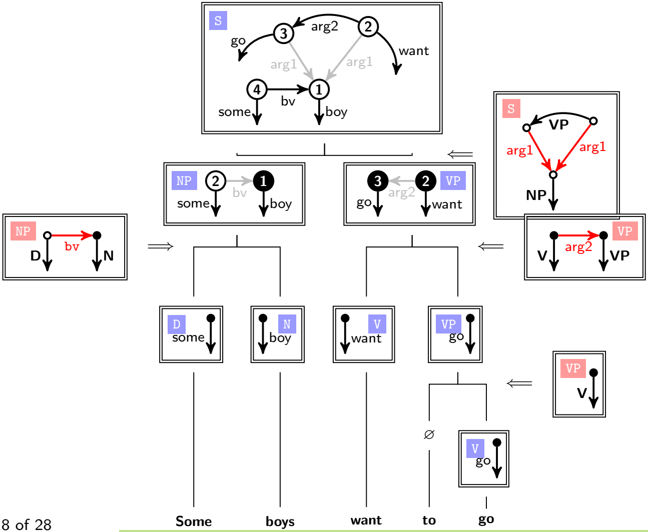
A top-down strategy



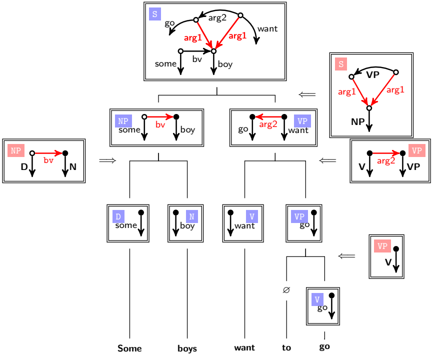
A top-down strategy



A top-down strategy

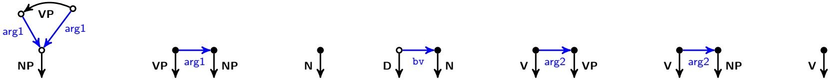


A top-down strategy



Key

At each step, we rely on some terminal edge(s) to identify applicable rules and thus decompose a sub-graph.



Regular graph grammar

Strong regularity

Sorcha Gilroy, Adam Lopez, Sebastian Maneth and Pijus Simonaitis. *(Re)introducing Regular Graph Languages*. 2017.

Weak regularity (our ongoing work)

A production rule is regular iff
every non-terminal edge of its right hand side is *anchored* by at least one terminal edge.

Practical graph-to-string parsing

(#n/#e)		Weak Regular	Strong Regular	Baseline
(12/23)	#subgraphs			26,414
	#total merge	565,222	4,422,904	4,878,124
	Time (s)	0.045	0.079	0.076
(16/23)	#subgraphs			53,965
	#total merge	1,694,389	21,176,306	23,478,324
	Time (s)	0.115	0.282	0.277
(20/42)	#subgraphs			71,261
	#total merge	1,654,275	39,131,493	41,291,199
	Time (s)	0.110	0.483	0.438
(23/45)	#subgraphs			188,961
	#total merge	79,648,439	1,056,812,108	1,089,545,027
	Time (s)	1.777	12.646	10.015
(28/59)	#subgraphs			297,708
	#total merge	466,191,707	7,971,458,311	8,032,173,533
	Time (s)	22.999	159.353	84.754

Practical graph-to-string parsing

(#n/#e)		Weak Regular	Strong Regular	Baseline
(12/23)	#subgraphs			26,414
	#total merge	565,222	4,422,904	4,878,124
	Time (s)	0.045	0.079	0.076
(16/23)	#subgraphs			53,965
	#total merge	1,694,389	21,176,306	23,478,324
	Time (s)	0.115	0.282	0.277
(20/42)	#subgraphs			71,261
	#total merge	1,654,275	39,131,493	41,291,199
	Time (s)	0.110	0.483	0.438
(23/45)	#subgraphs			188,961
	#total merge	79,648,439	1,056,812,108	1,089,545,027
	Time (s)	1.777	12.646	10.015
(28/59)	#subgraphs			297,708
	#total merge	466,191,707	7,971,458,311	8,032,173,533
	Time (s)	22.999	159.353	84.754

Practical graph-to-string parsing

(#n/#e)		Weak Regular	Strong Regular	Baseline
(12/23)	#subgraphs			26,414
	#total merge	565,222	4,422,904	4,878,124
	Time (s)	0.045	0.079	0.076
(16/23)	#subgraphs			53,965
	#total merge	1,694,389	21,176,306	23,478,324
	Time (s)	0.115	0.282	0.277
(20/42)	#subgraphs			71,261
	#total merge	1,654,275	39,131,493	41,291,199
	Time (s)	0.110	0.483	0.438
(23/45)	#subgraphs			188,961
	#total merge	79,648,439	1,056,812,108	1,089,545,027
	Time (s)	1.777	12.646	10.015
(28/59)	#subgraphs			297,708
	#total merge	466,191,707	7,971,458,311	8,032,173,533
	Time (s)	22.999	159.353	84.754

Practical graph-to-string parsing

(#n/#e)		Weak Regular	Strong Regular	Baseline
(12/23)	#subgraphs			26,414
	#total merge	565,222	4,422,904	4,878,124
	Time (s)	0.045	0.079	0.076
(16/23)	#subgraphs			53,965
	#total merge	1,694,389	21,176,306	23,478,324
	Time (s)	0.115	0.282	0.277
(20/42)	#subgraphs			71,261
	#total merge	1,654,275	39,131,493	41,291,199
	Time (s)	0.110	0.483	0.438
(23/45)	#subgraphs			188,961
	[illegible]	[illegible]	[illegible]	[illegible]
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
	#total merge	466,191,707	7,971,458,311	8,032,173,533
	Time (s)	22.999	159.353	84.754

Exact graph parsing can be practical.

Outline

Lexicalist vs. Constructivist

The recent study of events and argument structure in generative syntax, as pointed out by Marantz (2013), has shifted from the lexicalist approach to the constructivist approach.

- The interpretation of an event is determined by the syntactic configuration.
- The predicate only provides conceptual meaning.

Lexicalist approach	Constructivist approach
Chomsky (1970), Levin and Rappaport Hovav (1995)	Hale and Keyser (1993, 2002), Halle and Marantz (1993), Borer (2005a,b, 2013)
CCG, LFG, HPSG	Sign-Based Construction Grammar, Goldberg (1995, 2006)

Lexicalist vs. Constructivist

The recent study of events and argument structure in generative syntax, as pointed out by Marantz (2013), has shifted from the lexicalist approach to the constructivist approach.

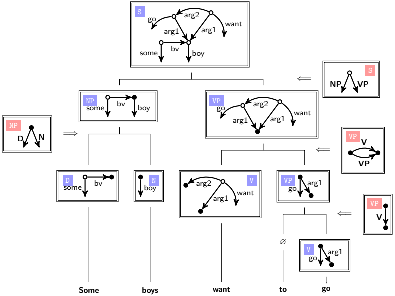
- The interpretation of an event is determined by the syntactic configuration.
- The predicate only provides conceptual meaning.

Lexicalist approach	Constructivist approach
Chomsky (1970), Levin and Rappaport Hovav (1995)	Hale and Keyser (1993, 2002), Halle and Marantz (1993), Borer (2005a,b, 2013)
CCG, LFG, HPSG	Sign-Based Construction Grammar, Goldberg (1995, 2006)

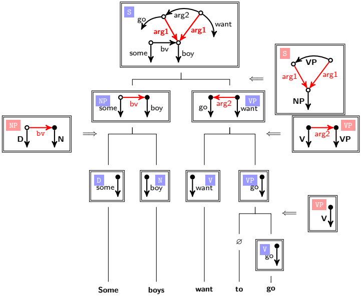
Lexicalist vs. Constructivist

- The interpretation of an event is determined by the syntactic configuration.
- The predicate only provides conceptual meaning.

Lexicalized Grammar

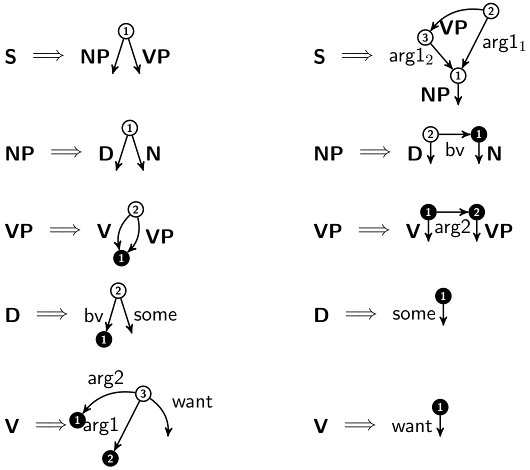


Construction Grammar



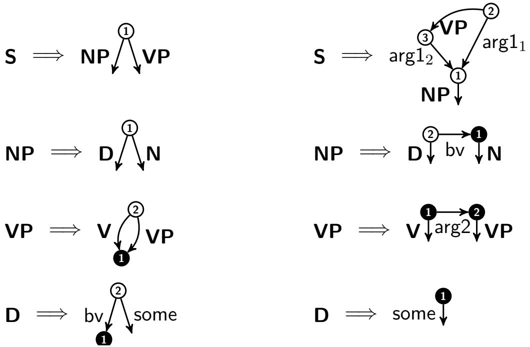
Lexicalist vs. Constructivist

Lexicalized Grammar   Construction Grammar



Lexicalist vs. Constructivist

Lexicalized Grammar | **Construction Grammar**



A significant number of production rules of any lexicalized grammar are not regular, but almost all production rules of a carefully designed construction grammar can be regular.

Constituency test

Replacement: If a group of words can be replaced with a single word,

Stand Alone: If a group of words can stand alone in response to a question,

Movement: If a group of words can be moved around in the sentence,

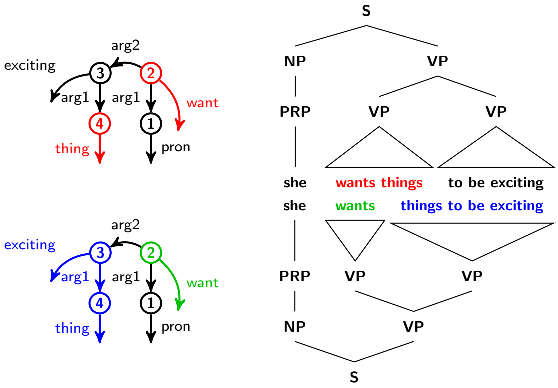
Coordination: If you can coordinate a group of words with a similar group of words,

Another perspective

> *By assuming incremental structure building it becomes possible to explain the differences between the range of constituents available to different diagnostics of constituency, including movement, ellipsis, coordination, scope and binding.*
>
> *Colin Phillips. Linear Order and Constituency.*

Constituency test

- Dana preferred for Pat to get the job.
- Could rising volatility possibly be ...
- ... with the additional $4.90 going to ...



Conclusion

1. How can we build a high-performance string-to-graph parser?

   ☺

2. How can we build a high-performance graph-to-string parser?

   ☺

3. Can we use a single model to achieve the two goals?

   ☺

4. Is our model linguistically meaningful?

   ☺

5. Can we apply our model to evaluate a linguistic hypothesis?

   ☺

Game over

Q What is the meaning of life?

A *life'*

Game over

Q What is the meaning of life?

A 

Game over

Q What is the meaning of life?

A 

Thank You!