Graph-Based Meaning Representations:

Design and Processing

Alexander Koller
Saarland University
koller@coli.uni-saarland.de

Stephan Oepen
University of Oslo
oe@ifi.uio.no

Weiwei Sun
Peking University
ws@pku.edu.cn

I reached into that funny little pocket that is high up on my dress.

$\xrightarrow{?}$ *Something is high up on my dress.*

Why Graph-Based Meaning Representation?

I reached into that funny little pocket that is high up on my dress.

$\longrightarrow$ *Something is high up on my dress.*

PETE

Why Graph-Based Meaning Representation?

I reached into that funny little pocket that is high up on my dress.

$\longrightarrow$ *Something is high up on my dress.*

PETE

A man, a woman and two girls are walking on the beach.

$\overset{?}{\longrightarrow}$ *A group of people is on a beach.*

SICK

Why Graph-Based Meaning Representation?

I reached into that funny little pocket that is high up on my dress.

$\longrightarrow$ *Something is high up on my dress.*

PETE

A man, a woman and two girls are walking on the beach.

$\longrightarrow$ *A group of people is on a beach.*

SICK

Why Graph-Based Meaning Representation?

I reached into that funny little pocket that is high up on my dress.

$\longrightarrow$ *Something is high up on my dress.*

PETE

A man, a woman and two girls are walking on the beach.

$\longrightarrow$ *A group of people is on a beach.*

SICK

The Commissioner doesn't regret that the President failed to make him leave Athens before May 2.

$\not\xrightarrow{?}$ *The Commissioner was in Athens on May 2.*

Cleo Condoravdi

Why Graph-Based Meaning Representation?

I reached into that funny little pocket that is high up on my dress.

$\longrightarrow$ *Something is high up on my dress.*

PETE

A man, a woman and two girls are walking on the beach.

$\longrightarrow$ *A group of people is on a beach.*

SICK

The Commissioner doesn't regret that the President failed to make him leave Athens before May 2.

$\stackrel{?}{\longrightarrow}$ *The Commissioner was in Athens on May 2.*

Cleo Condoravdi

Why Graph-Based Meaning Representation?

I reached into that funny little pocket that is high up on my dress.

$\longrightarrow$ *Something is high up on my dress.*

PETE

A man, a woman and two girls are walking on the beach.

$\longrightarrow$ *A group of people is on a beach.*

SICK

The Commissioner doesn't regret that the President failed to make him leave Athens before May 2.

$\overset{?}{\longrightarrow}$ *The Commissioner was in Athens on May 2.*

Cleo Condoravdi

Entailment, for Example

- What types of information are at play in reasoning about entailment?
- Who did what to whom, when and where? Reference, veridicality, etc.

Why Graph-Based Meaning Representation?

I reached into that funny little pocket that is high up on my dress.

$\longrightarrow$ *Something is high up on my dress.*

PETE

A man, a woman and two girls are walking on the beach.

$\longrightarrow$ *A group of people is on a beach.*

SICK

The Commissioner doesn't regret that the President failed to make him leave Athens before May 2.

$\stackrel{?}{\not\longrightarrow}$ *The Commissioner was in Athens on May 2.*

Cleo Condoravdi

Entailment, for Example

- What types of information are at play in reasoning about entailment?
- Who did what to whom, when and where? Reference, veridicality, etc.

$\rightarrow$ Logical inference or distributional approximation, both need structure.

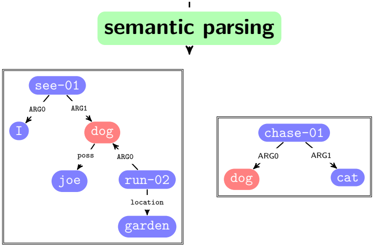
Preview: Semantic Structure in Applications

I saw Joe's dog, which was running in the garden.
The dog was chasing a cat.

(Liu et al., 2015; Hardy & Vlachos, 2018)

Preview: Semantic Structure in Applications

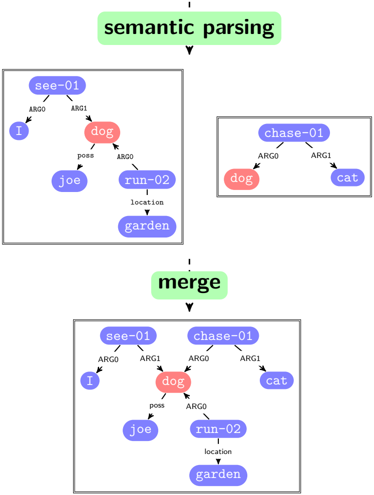
I saw Joe's dog, which was running in the garden.
The dog was chasing a cat.



(Liu et al., 2015; Hardy & Vlachos, 2018)

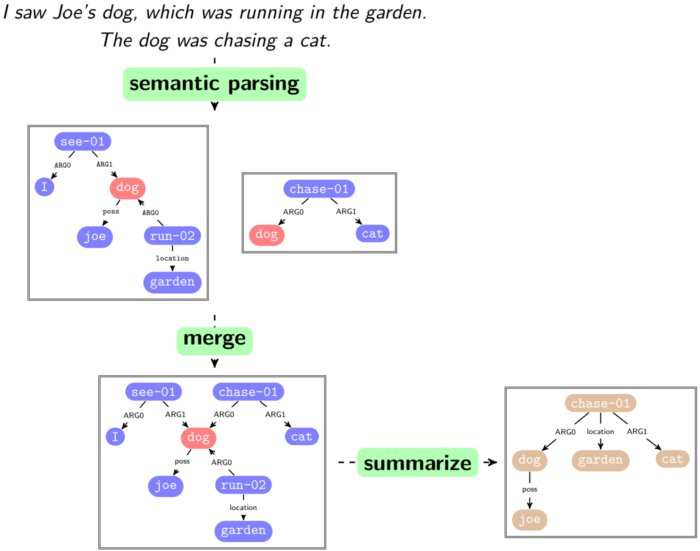
Preview: Semantic Structure in Applications

I saw Joe's dog, which was running in the garden.
The dog was chasing a cat.



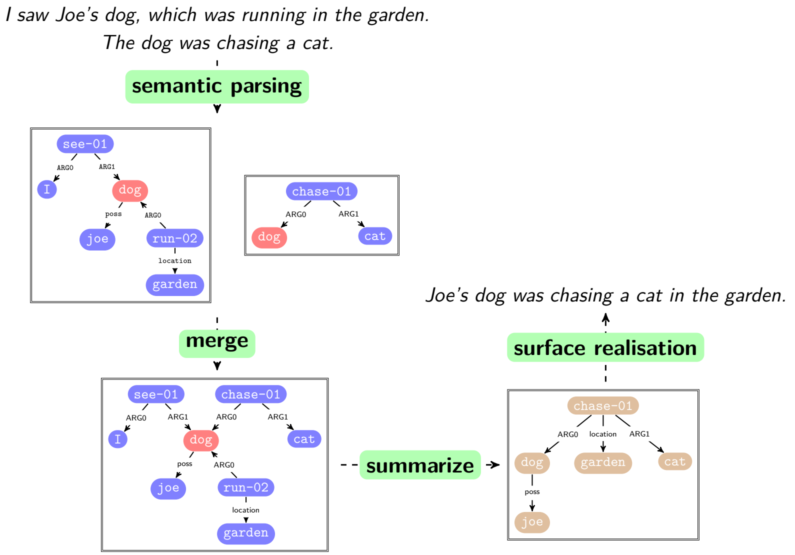
(Liu et al., 2015; Hardy & Vlachos, 2018)

Preview: Semantic Structure in Applications



(Liu et al., 2015; Hardy & Vlachos, 2018)

Preview: Semantic Structure in Applications



(Liu et al., 2015; Hardy & Vlachos, 2018)

High-Level Goals of the Tutorial

Graph-Based Representations of Meaning

- Vast, complex landscape of representing natural language meaning;
- diverse linguistic traditions, modeling assumptions, levels of ambition;
- some differences are superficial (e.g. terminology), others run deeper;

$\rightarrow$ clarify concepts and vocabulary; high-level survey of selected resources.

High-Level Goals of the Tutorial

Graph-Based Representations of Meaning

- Vast, complex landscape of representing natural language meaning;
- diverse linguistic traditions, modeling assumptions, levels of ambition;
- some differences are superficial (e.g. terminology), others run deeper;

$\rightarrow$ clarify concepts and vocabulary; high-level survey of selected resources.

Parsing into Graph-Structured Representations

- Cottage industry of parsers with outputs structures beyond rooted trees;
- distinct techniques, e.g. based on transitions, composition, 'translation';
- some framework-internal evolution: design reflects specific assumptions;

$\rightarrow$ tease apart sub-tasks and families of approaches; review representatives.

High-Level Goals of the Tutorial

Graph-Based Representations of Meaning

- Vast, complex landscape of representing natural language meaning;
- diverse linguistic traditions, modeling assumptions, levels of ambition;
- some differences are superficial (e.g. terminology), others run deeper;

→ clarify concepts and vocabulary; high-level survey of selected resources.

Parsing into Graph-Structured Representations

- Cottage industry of parsers with outputs structures beyond rooted trees;
- distinct techniques, e.g. based on transitions, composition, ‘translation’;
- some framework-internal evolution: design reflects specific assumptions;

→ tease apart sub-tasks and families of approaches; review representatives.

Fragmentation and ‘Balkanization’

- Cross-Framework Perspective: Seek commonality and complementarity.

Outline: Our Game Plan

(1) & (2) Foundations: Linguistic & Formal (0:30)

- Tease apart various ‘facets’ (layers) of meaning; common terminology.

(3) Graph-Based Meaning Banks (0:45)

- Selective, dense review of five English corpora with semantic graphs;
- highlight distinct design decisions and goals; contrast across schools.

(4) Parsing into Semantic Graphs (1:00)

- Factorization-, composition-, transition-, translation-based techniques;
- graph similarity evaluation; cross-framework and cross-lingual parsing.

(5) & (6) Outlook: Using Semantic Graphs (0:15)

- Example use cases: summarization, entity linking, machine translation.

Outline: Our Game Plan

(1) & (2) Foundations: Linguistic & Formal (0:30)

- Tease apart various ‘facets’ (layers) of meaning; common terminology.

(3) Graph-Based Meaning Banks (0:45)

- Selective, dense review of five English corpora with semantic graphs;
- highlight distinct design decisions and goals; contrast across schools.

(4) Parsing into Semantic Graphs (1:00)

- Factorization-, composition-, transition-, translation-based techniques;
- graph similarity evaluation; cross-framework and cross-lingual parsing.

(5) & (6) Outlook: Using Semantic Graphs (0:15)

- Example use cases: summarization, entity linking, machine translation.

(Some) Time for Questions at the End of Each Block

1 Foundations: Semantics

What do we Mean by ‘Meaning’?

Abrams gave Browne a book.

Abrams gave a book to Browne.

Browne was given a book by Abrams.

A book was given to Browne by Abrams.

Browne, Abrams gave the book to.

A book, Browne was given by Abrams.

. . .

What do we Mean by ‘Meaning’?

Abrams gave Browne a book.

Abrams gave a book to Browne.

Browne was given a book by Abrams.

A book was given to Browne by Abrams.

Browne, Abrams gave the book to.

A book, Browne was given by Abrams.

...

The question is difficult to answer precisely.

It is difficult to answer the question precisely.

What do we Mean by ‘Meaning’?

Abrams gave Browne a book.

Abrams gave a book to Browne.

Browne was given a book by Abrams.

A book was given to Browne by Abrams.

Browne, Abrams gave the book to.

A book, Browne was given by Abrams.

...

The question is difficult to answer precisely.

It is difficult to answer the question precisely.

the patient’s arrival	*the arrival by the patient*
the office manager	*the manager of the office*

What do we Mean by ‘Meaning’?

Abrams gave Browne a book.
Abrams gave a book to Browne.
Browne was given a book by Abrams.
A book was given to Browne by Abrams.
Browne, Abrams gave the book to.
A book, Browne was given by Abrams.
...

The question is difficult to answer precisely.
It is difficult to answer the question precisely.

the patient’s arrival	*the arrival by the patient*
the office manager	*the manager of the office*

- Superficially different linguistic forms can describe the same situation;
- hold true under the same circumstances; can substitute for each other;

What do we Mean by ‘Meaning’?

Abrams gave Browne a book.

Abrams gave a book to Browne.

Browne was given a book by Abrams.

A book was given to Browne by Abrams.

Browne, Abrams gave the book to.

A book, Browne was given by Abrams.

...

The question is difficult to answer precisely.

It is difficult to answer the question precisely.

the patient’s arrival	*the arrival by the patient*
the office manager	*the manager of the office*

- Superficially different linguistic forms can describe the same situation;
- hold true under the same circumstances; can substitute for each other;

→ close paraphrases: convey the ‘same meaning’ (in unmarked contexts).

Sentence vs. Speaker Meaning

Linguistic semantics does not furnish a characterization of the interpretation of utterances in use, which is what one finally needs for natural language understanding applications—rather, it (mostly) provides a characterization of conventional content, that part of meaning determined by linguistic form. [...] In order to interpret correctly, one must exploit further knowledge sources [...]: domain knowledge, common sense, communicative purpose, extralinguistic tasks, assumptions of interlocutors about each other. (Nerbonne, 1994, p. 134)

Sentence vs. Speaker Meaning

Linguistic semantics does not furnish a characterization of the interpretation of utterances in use, which is what one finally needs for natural language understanding applications—rather, it (mostly) provides a characterization of conventional content, that part of meaning determined by linguistic form. [...] In order to interpret correctly, one must exploit further knowledge sources [...]: domain knowledge, common sense, communicative purpose, extralinguistic tasks, assumptions of interlocutors about each other. (Nerbonne, 1994, p. 134)

Common Distinction in Linguistic Semantics—Challenging to Make Precise

He wants to be seen by her. vs. *He wants her to see him.*

Sentence vs. Speaker Meaning

> *Linguistic semantics does not furnish a characterization of the interpretation of utterances in use, which is what one finally needs for natural language understanding applications—rather, it (mostly) provides a characterization of conventional content, that part of meaning determined by linguistic form. [...] In order to interpret correctly, one must exploit further knowledge sources [...]: domain knowledge, common sense, communicative purpose, extralinguistic tasks, assumptions of interlocutors about each other.* (Nerbonne, 1994, p. 134)

Common Distinction in Linguistic Semantics—Challenging to Make Precise

He wants to be seen by her. vs. *He wants her to see him.*

Han	*elsker*	*sin*	*hatt.*	vs.	*Han*	*elsker*	*hans*	*hatt.*
He_i	loves	his_i	hat.		He_i	loves	$\text{his}_{j \neq i}$	hat.

Sentence vs. Speaker Meaning

Linguistic semantics does not furnish a characterization of the interpretation of utterances in use, which is what one finally needs for natural language understanding applications—rather, it (mostly) provides a characterization of conventional content, that part of meaning determined by linguistic form. [...] In order to interpret correctly, one must exploit further knowledge sources [...]: domain knowledge, common sense, communicative purpose, extralinguistic tasks, assumptions of interlocutors about each other. (Nerbonne, 1994, p. 134)

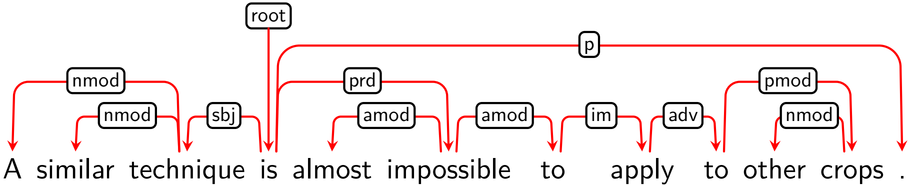
Common Distinction in Linguistic Semantics—Challenging to Make Precise

He wants to be seen by her. vs. *He wants her to see him.*

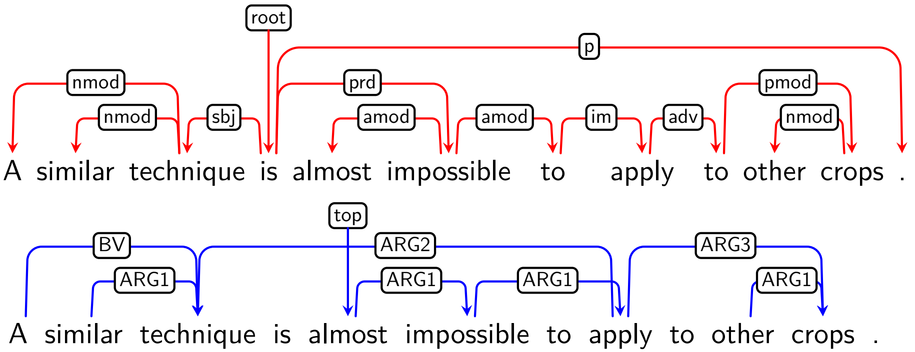
Han	*elsker*	*sin*	*hatt.*	vs.	*Han*	*elsker*	*hans*	*hatt.*
He$_i$	loves	his$_i$	hat.		He$_i$	loves	his$_{j \neq i}$	hat.

Sherlock saw the suspect with the binoculars.

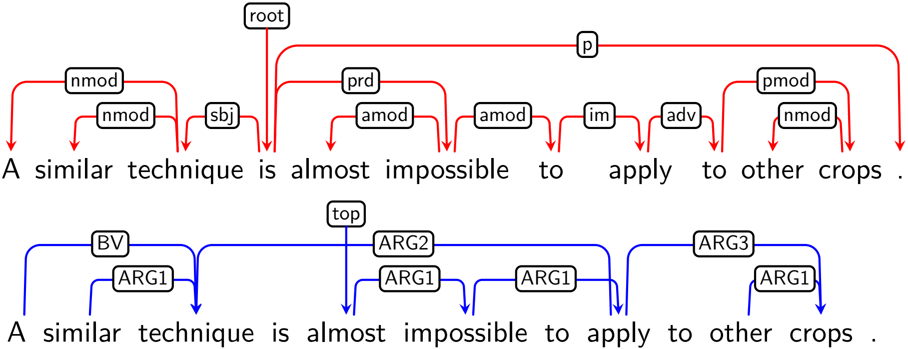
Syntactic Trees vs. Semantic Graphs [WSJ #0209013]



Syntactic Trees vs. Semantic Graphs [WSJ #0209013]



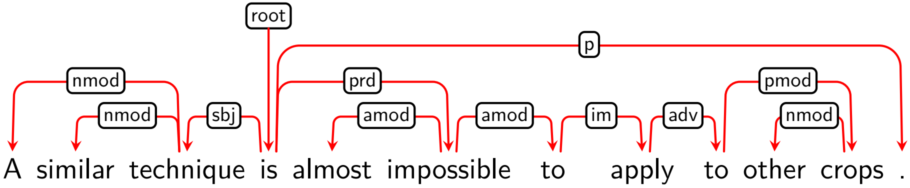
Syntactic Trees vs. Semantic Graphs [WSJ #0209013]

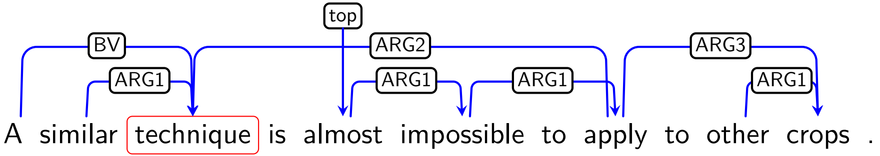


$$\exists x : \text{technique'}(x) \wedge \text{similar'}(x), \exists y : \text{crop'}(y) \wedge \text{other'}(y)$$

$$\rightarrow \text{almost'}(\neg\text{possible'}(\text{apply'}(\_, x, y)))$$

Syntactic Trees vs. Semantic Graphs [WSJ #0209013]

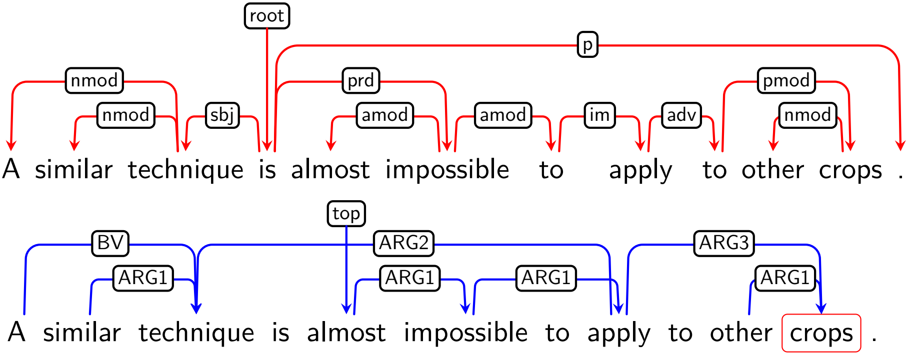




$$\exists {\color{red}x} : \text{technique'}({\color{red}x}) \land \text{similar'}({\color{red}x}), \exists y : \text{crop'}(y) \land \text{other'}(y)$$

$$\rightarrow \text{almost'}(\neg\text{possible'}(\text{apply'}(\_, {\color{red}x}, y)))$$

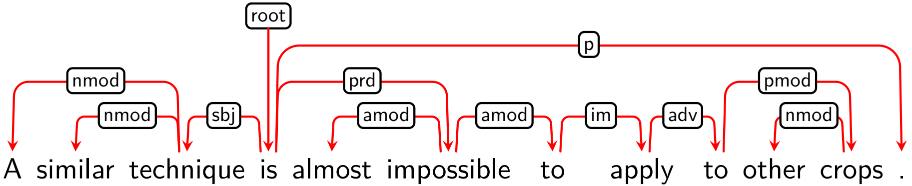
Syntactic Trees vs. Semantic Graphs [WSJ #0209013]

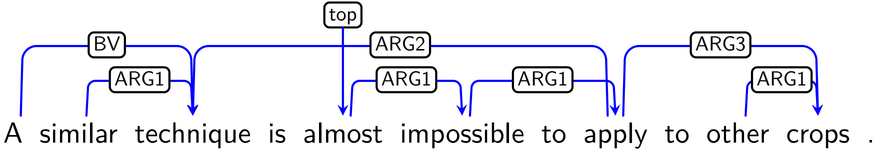


$$\exists x : \text{technique'}(x) \wedge \text{similar'}(x), \exists {\color{red}y} : \text{crop'}({\color{red}y}) \wedge \text{other'}({\color{red}y})$$

$$\rightarrow \text{almost'}(\neg\text{possible'}(\text{apply'}(\_, x, {\color{red}y})))$$

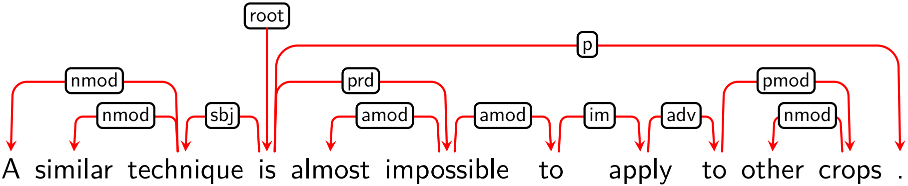
Syntactic Trees vs. Semantic Graphs [WSJ #0209013]

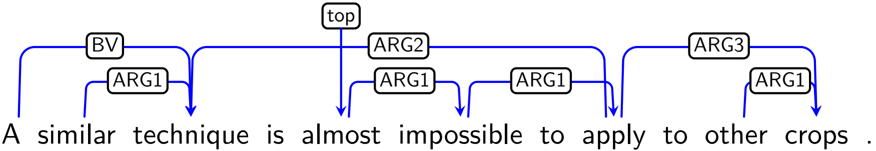




$$\exists x : \text{technique'}(x) \wedge \text{similar'}(x), \exists y : \text{crop'}(y) \wedge \text{other'}(y)$$
$$\rightarrow \text{almost'}(\neg\text{possible'}(\text{apply'}(\_, x, y)))$$

Syntactic Trees vs. Semantic Graphs [WSJ #0209013]

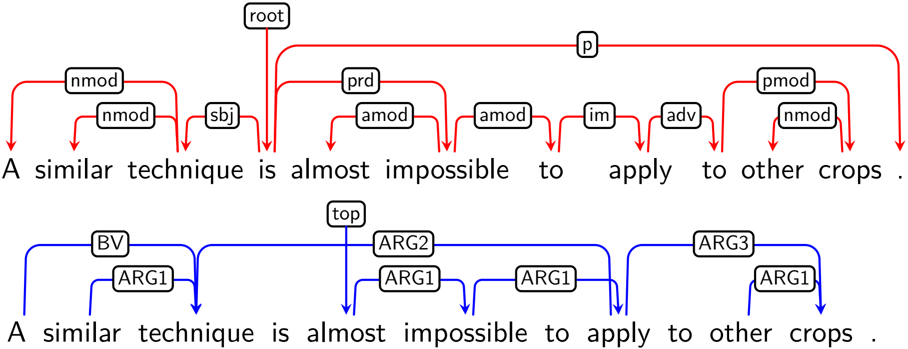




$$\exists x : \text{technique}'(x) \land \text{similar}'(x, \_), \exists y : \text{crop}'(y) \land \text{other}'(y, \_)$$

$$\rightarrow \text{almost}'(\neg\text{possible}'(\text{apply}'(\_, x, y)))$$

Syntactic Trees vs. Semantic Graphs [WSJ #0209013]



$$\exists x : \text{technique'}(x) \wedge \text{similar'}(x, \_), \exists y : \text{crop'}(y) \wedge \text{other'}(y, \_)$$

$$\rightarrow \text{almost'}(\neg\text{possible'}(\text{apply'}(\_, x, y)))$$

Different Desiderata (and Levels of Abstraction)

- Grammaticality (e.g. subject–verb agreement) vs. relational structure.

Semi-Formally: Trees vs. Graphs

Structural Wellformedness Conditions on Trees

- Unique root, connected, single parent, free of cycles; maybe: projective;

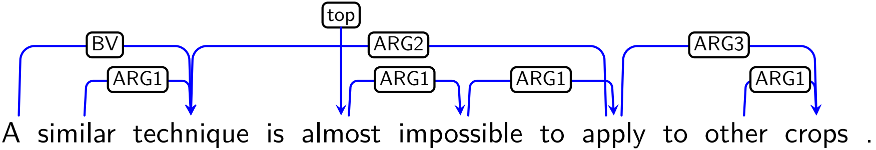
$\rightarrow$ all nodes (but the root) reachable by unique directed path from root.

Semi-Formally: Trees vs. Graphs

Structural Wellformedness Conditions on Trees

- Unique root, connected, single parent, free of cycles; maybe: projective;

→ all nodes (but the root) reachable by unique directed path from root.

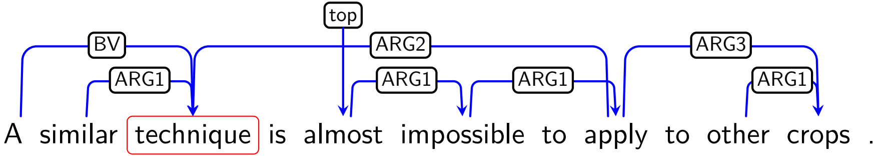


Semi-Formally: Trees vs. Graphs

Structural Wellformedness Conditions on Trees

- Unique root, connected, single parent, free of cycles; maybe: projective;

→ all nodes (but the root) reachable by unique directed path from root.



Beyond Trees: General Graphs

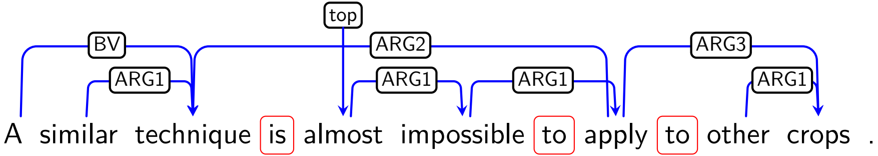
- Argument sharing: nodes with multiple incoming edges (***in***-degree > 1);

Semi-Formally: Trees vs. Graphs

Structural Wellformedness Conditions on Trees

- Unique root, connected, single parent, free of cycles; maybe: projective;

→ all nodes (but the root) reachable by unique directed path from root.



Beyond Trees: General Graphs

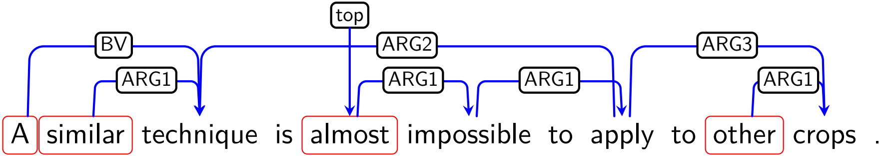
- Argument sharing: nodes with multiple incoming edges (***in***-degree > 1);
- some surface tokens do not contribute meaning (many function words);

Semi-Formally: Trees vs. Graphs

Structural Wellformedness Conditions on Trees

- Unique root, connected, single parent, free of cycles; maybe: projective;

$\rightarrow$ all nodes (but the root) reachable by unique directed path from root.



Beyond Trees: General Graphs

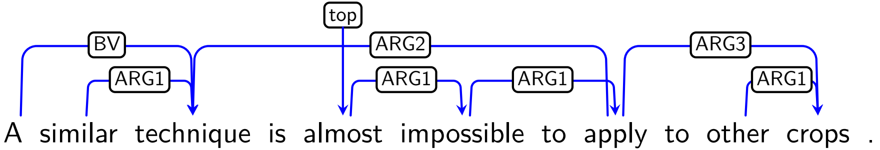
- Argument sharing: nodes with multiple incoming edges (***in***-degree > 1);
- some surface tokens do not contribute meaning (many function words);
- (structurally) multi-rooted: more than one node with zero ***in***-degree;

Semi-Formally: Trees vs. Graphs

Structural Wellformedness Conditions on Trees

- Unique root, connected, single parent, free of cycles; maybe: projective;

→ all nodes (but the root) reachable by unique directed path from root.



Beyond Trees: General Graphs

- Argument sharing: nodes with multiple incoming edges (***in***-degree > 1);
- some surface tokens do not contribute meaning (many function words);
- (structurally) multi-rooted: more than one node with zero ***in***-degree;

→ massive growth in modeling and algorithmic complexity (NP-complete).

Terminology: Syntactic vs. Semantic ‘Predicates’

Syntax: Head–Dependent Relations

- Heads license and govern, e.g. verbs, relational nouns, prepositions;
- complements include subjects, objects, obliques, clausal arguments;
- adjuncts add information, e.g. adjectives, adverbials, relative clauses:

Abrams bet $10 on Monday that it rained in Florence.

Terminology: Syntactic vs. Semantic ‘Predicates’

Syntax: Head–Dependent Relations

- Heads license and govern, e.g. verbs, relational nouns, prepositions;
- complements include subjects, objects, obliques, clausal arguments;
- adjuncts add information, e.g. adjectives, adverbials, relative clauses:

Abrams bet $10 on Monday that it rained in Florence.

Terminology: Syntactic vs. Semantic ‘Predicates’

Syntax: Head–Dependent Relations

- Heads license and govern, e.g. verbs, relational nouns, prepositions;
- complements include subjects, objects, obliques, clausal arguments;
- adjuncts add information, e.g. adjectives, adverbials, relative clauses:

Abrams bet $10 on Monday that it rained in Florence.

Terminology: Syntactic vs. Semantic 'Predicates'

Syntax: Head–Dependent Relations

- Heads license and govern, e.g. verbs, relational nouns, prepositions;
- complements include subjects, objects, obliques, clausal arguments;
- adjuncts add information, e.g. adjectives, adverbials, relative clauses:

Abrams bet $10 on Monday that it rained in Florence.

Terminology: Syntactic vs. Semantic ‘Predicates’

Syntax: Head–Dependent Relations

- Heads license and govern, e.g. verbs, relational nouns, prepositions;
- complements include subjects, objects, obliques, clausal arguments;
- adjuncts add information, e.g. adjectives, adverbials, relative clauses:

Abrams bet $10 on Monday that it rained in Florence.

Semantics: Predicate–Argument Relations

- Predicates evoke relations of variable arity; from all major word classes;
- arguments fill semantic roles, defined relative to a specific predicate;

Terminology: Syntactic vs. Semantic 'Predicates'

Syntax: Head–Dependent Relations

- Heads license and govern, e.g. verbs, relational nouns, prepositions;
- complements include subjects, objects, obliques, clausal arguments;
- adjuncts add information, e.g. adjectives, adverbials, relative clauses:

Abrams bet $10 on Monday that it rained in Florence.

Semantics: Predicate–Argument Relations

- Predicates evoke relations of variable arity; from all major word classes;
- arguments fill semantic roles, defined relative to a specific predicate;

Abrams bet $10 on Monday that it rained in Florence.

Terminology: Syntactic vs. Semantic 'Predicates'

Syntax: Head–Dependent Relations

- Heads license and govern, e.g. verbs, relational nouns, prepositions;
- complements include subjects, objects, obliques, clausal arguments;
- adjuncts add information, e.g. adjectives, adverbials, relative clauses:

Abrams bet $10 on Monday that it rained in Florence.

Semantics: Predicate–Argument Relations

- Predicates evoke relations of variable arity; from all major word classes;
- arguments fill semantic roles, defined relative to a specific predicate;

Abrams bet $10 on Monday that it rained in Florence.

Terminology: Syntactic vs. Semantic 'Predicates'

Syntax: Head–Dependent Relations

- Heads license and govern, e.g. verbs, relational nouns, prepositions;
- complements include subjects, objects, obliques, clausal arguments;
- adjuncts add information, e.g. adjectives, adverbials, relative clauses:

Abrams bet $10 on Monday that it rained in Florence.

Semantics: Predicate–Argument Relations

- Predicates evoke relations of variable arity; from all major word classes;
- arguments fill semantic roles, defined relative to a specific predicate;

Abrams bet $10 on Monday that it rained in Florence.

Terminology: Syntactic vs. Semantic ‘Predicates’

Syntax: Head–Dependent Relations

- Heads license and govern, e.g. verbs, relational nouns, prepositions;
- complements include subjects, objects, obliques, clausal arguments;
- adjuncts add information, e.g. adjectives, adverbials, relative clauses:

Abrams bet $10 on Monday that it rained in Florence.

Semantics: Predicate–Argument Relations

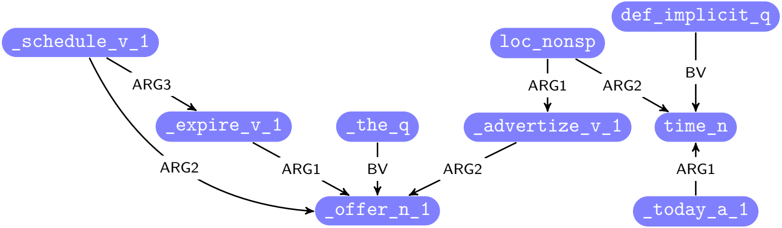
- Predicates evoke relations of variable arity; from all major word classes;
- arguments fill semantic roles, defined relative to a specific predicate;

Abrams bet $10 on Monday that it rained in Florence.

- Prepositions *on* or *in* as two-place relations, e.g. temporal or locative.

Reflections on Predicate–Argument Structure

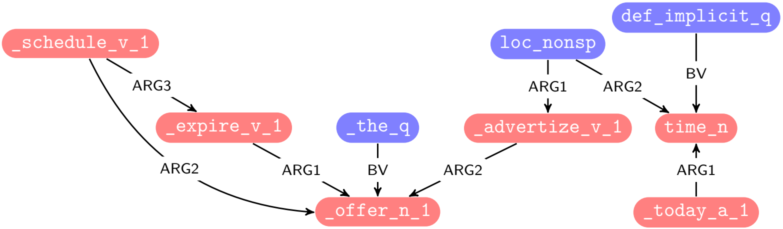
- Arguments recursively are predicates most of the time (and vice versa);



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

Reflections on Predicate–Argument Structure

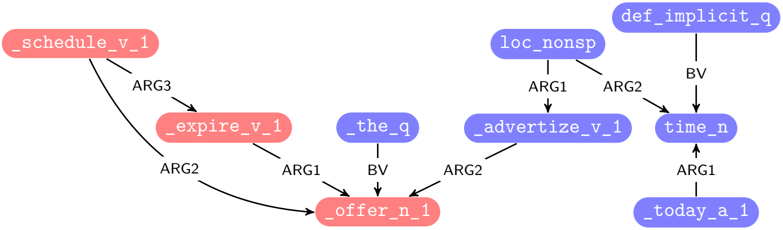
- Arguments recursively are predicates most of the time (and vice versa);
- 'content words' introduce predicates, e.g. nouns, verbs, and adjectives;



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

- Arguments recursively are predicates most of the time (and vice versa);
- 'content words' introduce predicates, e.g. nouns, verbs, and adjectives;
- arity determines the (possible) number of arguments: unary, binary, . . . ;
- seemingly zero-place predicates can have referential argument (ARG0):

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



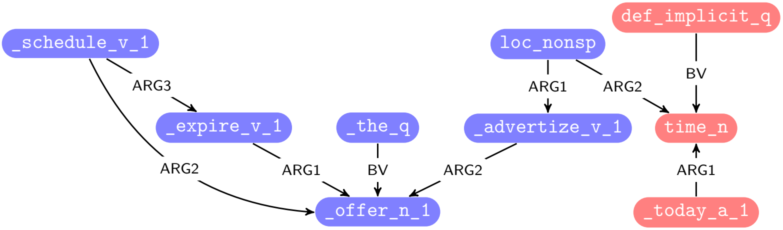
The offer advertized today is scheduled to expire. [WSJ #0032002]

Reflections on Predicate–Argument Structure

- Arguments recursively are predicates most of the time (and vice versa);
- ‘content words’ introduce predicates, e.g. nouns, verbs, and adjectives;
- arity determines the (possible) number of arguments: unary, binary, . . . ;
- seemingly zero-place predicates can have referential argument (ARG0):

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

- decomposition: possibly multiple predicates per word or construction.



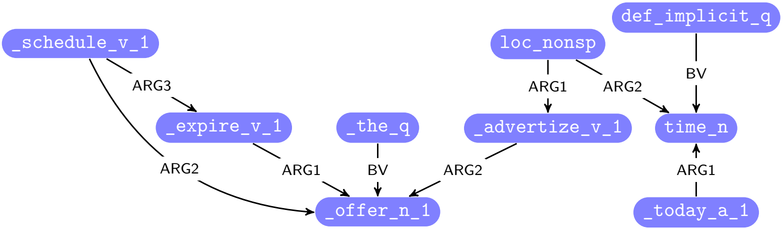
The offer advertized today is scheduled to expire. [WSJ #0032002]

Reflections on Predicate–Argument Structure

- Arguments recursively are predicates most of the time (and vice versa);
- 'content words' introduce predicates, e.g. nouns, verbs, and adjectives;
- arity determines the (possible) number of arguments: unary, binary, . . . ;
- seemingly zero-place predicates can have referential argument (ARG0):

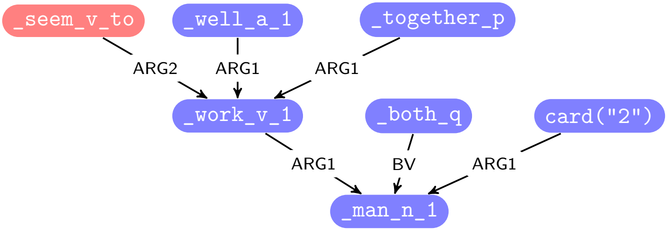
$$\text{offer'}(x) \wedge \text{schedule'}(\_, x, \text{expire'}(x))$$

- decomposition: possibly multiple predicates per word or construction.



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

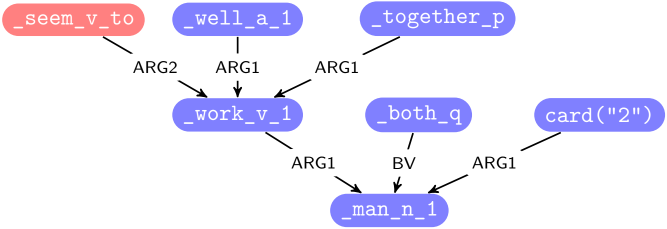
Mismatches between Syntax and Semantics



Both men seem to work well together. [WSJ #0109043]

- Discord between head–dependent and predicate–argument relations;
- syntactic subject of *seem* is a semantic argument of its complement;

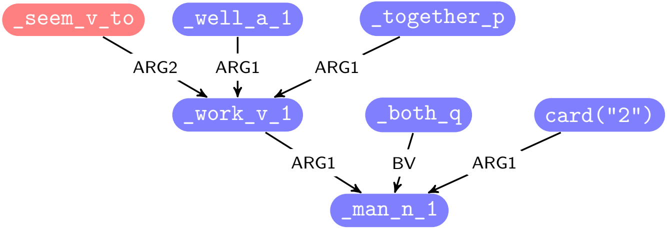
Mismatches between Syntax and Semantics



Both men seem to work well together. [WSJ #0109043]

- Discord between head–dependent and predicate–argument relations;
- syntactic subject of *seem* is a semantic argument of its complement;
- consider close paraphrase: *It seems that both men work well together.*
- expletive *it* subject is not referential, hence no semantic contribution;

Mismatches between Syntax and Semantics



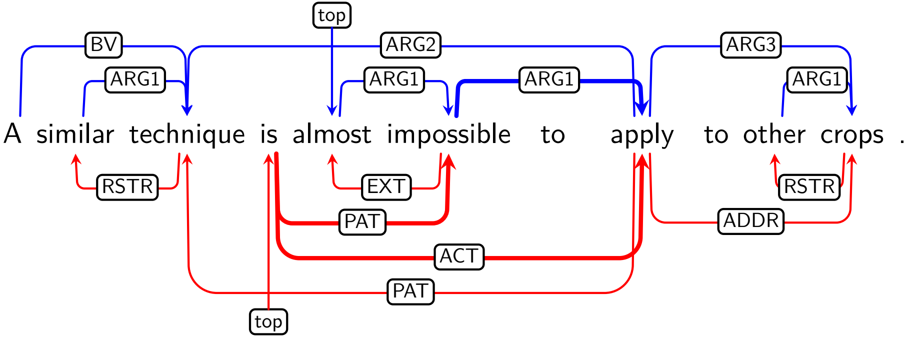
Both men seem to work well together. [WSJ #0109043]

- Discord between head–dependent and predicate–argument relations;
- syntactic subject of *seem* is a semantic argument of its complement;
- consider close paraphrase: *It seems that both men work well together.*
- expletive *it* subject is not referential, hence no semantic contribution;
- about two dozen subject raising verbs in broad-coverage English lexicon.

- There is much variation in analysis of individual linguistic phenomena;
- specific semantic framework requires many interacting design decisions;

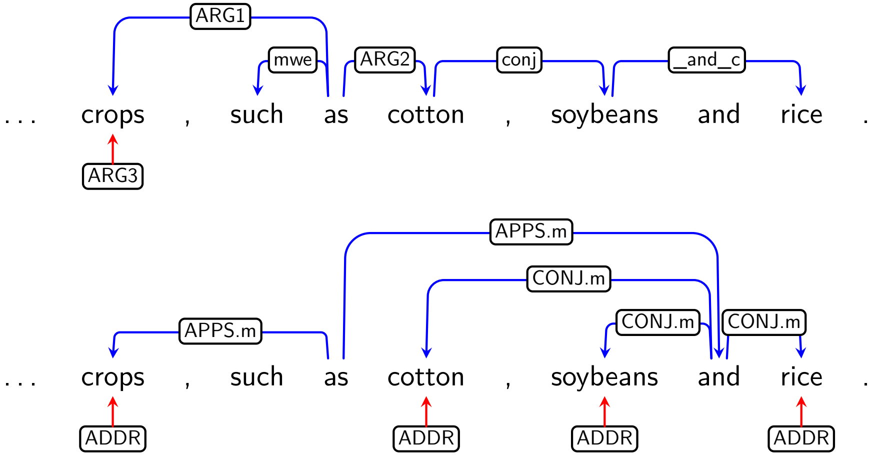
Example Design Decisions: Copula Constructions

- There is much variation in analysis of individual linguistic phenomena;
- specific semantic framework requires many interacting design decisions;
- divergent views already at the level of which words are content-bearing;
- for example, the predicative copula: *the fierce dog* vs. *the dog is fierce*.



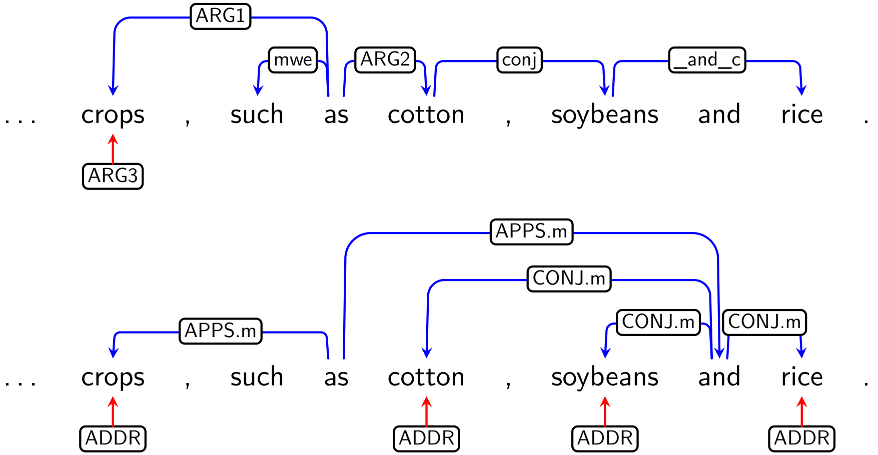
Example Design Decisions: Coordination

- No less frequent: coordinate structures (sometimes called parataxis);
- different types: *Abrams arrived and ate* vs. *Abrams and Brown arrived.*



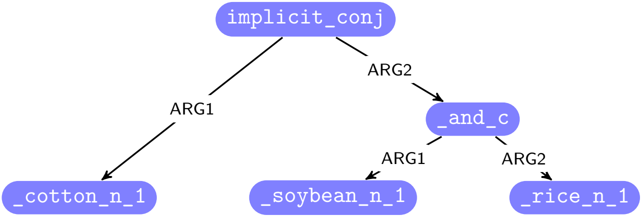
Example Design Decisions: Coordination

- No less frequent: coordinate structures (sometimes called parataxis);
- different types: *Abrams arrived and ate* vs. *Abrams and Brown arrived.*
- intuitively, a kind of 'grouping'; how to represent the group as a whole?



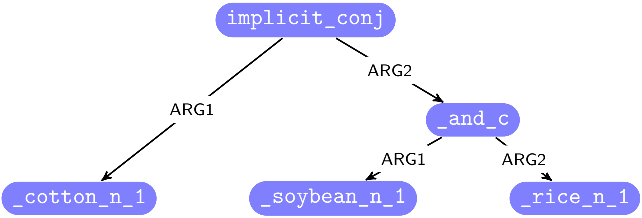
Further Variations on Coordination

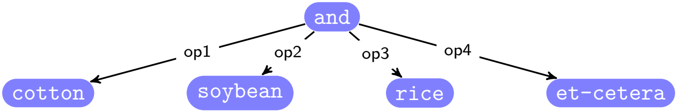
cotton, soybeans, and rice



Further Variations on Coordination

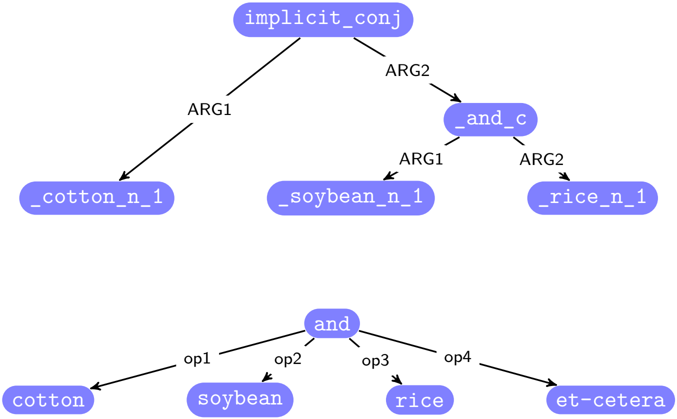
cotton, soybeans, and rice





Further Variations on Coordination

cotton, soybeans, and rice



Popel et al. (2013): nine distinct syntactico-semantic dependency patterns.

Background: Partial Predicate–Argument Structures

Since Late 1990s, Semantic Role Labeling (SRL)

- Predicate–argument structure with strong focus on verbal predicates;
- large, manually annotated corpora available (for multiple languages), e.g. Framenet (Baker et al., 1998), PropBank (Palmer et al., 2005);

Background: Partial Predicate–Argument Structures

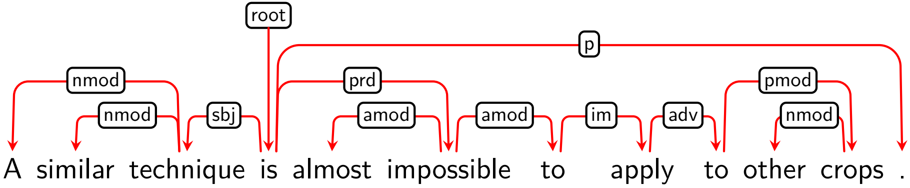
Since Late 1990s, Semantic Role Labeling (SRL)

- Predicate–argument structure with strong focus on verbal predicates;
- large, manually annotated corpora available (for multiple languages), e.g. Framenet (Baker et al., 1998), PropBank (Palmer et al., 2005);
- argument entities originally syntactic constituents or simply text spans;

Background: Partial Predicate–Argument Structures

Since Late 1990s, Semantic Role Labeling (SRL)

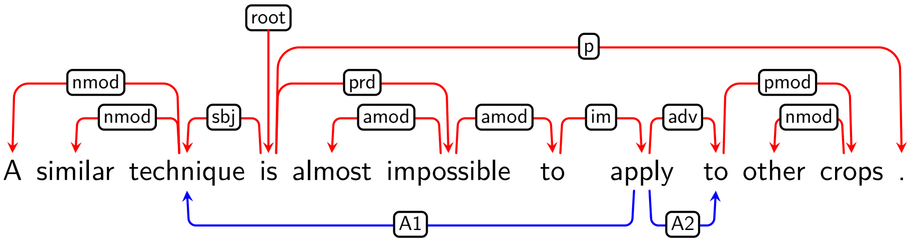
- Predicate–argument structure with strong focus on verbal predicates;
- large, manually annotated corpora available (for multiple languages), e.g. Framenet (Baker et al., 1998), PropBank (Palmer et al., 2005);
- argument entities originally syntactic constituents or simply text spans;
- 2008 Shared Task on joint syntactic and semantic dependency parsing:



Background: Partial Predicate–Argument Structures

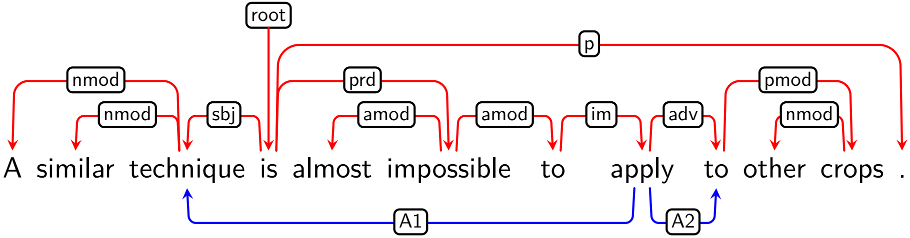
Since Late 1990s, Semantic Role Labeling (SRL)

- Predicate–argument structure with strong focus on verbal predicates;
- large, manually annotated corpora available (for multiple languages), e.g. Framenet (Baker et al., 1998), PropBank (Palmer et al., 2005);
- argument entities originally syntactic constituents or simply text spans;
- 2008 Shared Task on joint syntactic and semantic dependency parsing:



Since Late 1990s, Semantic Role Labeling (SRL)

- Predicate–argument structure with strong focus on verbal predicates;
- large, manually annotated corpora available (for multiple languages), e.g. Framenet (Baker et al., 1998), PropBank (Palmer et al., 2005);
- argument entities originally syntactic constituents or simply text spans;
- 2008 Shared Task on joint syntactic and semantic dependency parsing:

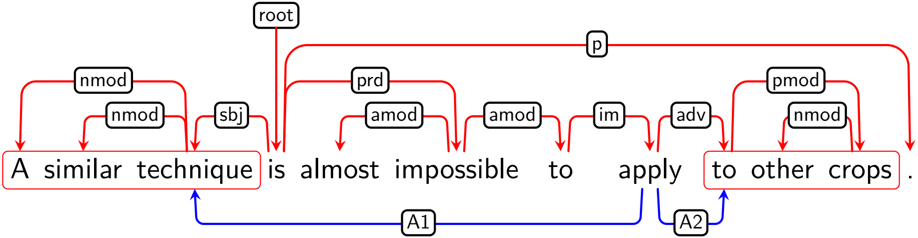


- Several (semantic) predicate–argument relations remain unannotated;

Background: Partial Predicate–Argument Structures

Since Late 1990s, Semantic Role Labeling (SRL)

- Predicate–argument structure with strong focus on verbal predicates;
- large, manually annotated corpora available (for multiple languages), e.g. Framenet (Baker et al., 1998), PropBank (Palmer et al., 2005);
- argument entities originally syntactic constituents or simply text spans;
- 2008 Shared Task on joint syntactic and semantic dependency parsing:



- Several (semantic) predicate–argument relations remain unannotated;
- conversion to bilexical dependency graphs, head selection from syntax.

Beyond Predicate–Argument Structure

Many NL expressions add logical structure
on top of the predicate-argument structure.

Negation

- John does not eat cookies.
- John said that Mary does not like cookies.
  vs. John did not say that Mary likes cookies.

Quantification

- Every boy likes cookies.
- Every boy ate a cookie.
- *All funny jokes are short.* vs. *All short jokes are funny.*
- Israel stood still as eight soldiers from all branches of the military carried the coffin to the burial ground.

FraCaS: An Early 'Meaning Bank'

Fracas #074

P1	All/most Europeans can travel freely within Europe.
Q	Can all/most Europeans who reside outside of Europe travel freely within Europe?
H	All/most Europeans who reside outside of Europe can travel freely within Europe.

- EU project on computational semantics in the mid-1990s.
- Collected 346 (non-)entailment sentence tuples.
- Also annotated with popular semantic representations of the time (predicate logic, DRT, etc.); but the annotations were lost.

Presupposition and Focus

Certain semantic phenomena supply meaning beyond the truth conditions of the sentence.

Presupposition

- A presupposition of a sentence is a piece of meaning that survives even if the sentence is negated.
- Today I took my cat to the vet.
  Today I didn't take my cat to the vet.
  Did you take your cat to the vet today?

Focus

- A focused phrase implicitly evokes alternatives of which the predication is false.
- YOUR children don't hate school.
  Your CHILDREN don't hate school. . . .

Word Senses

The words in a sentence may be ambiguous with respect to their senses. The semantic annotation may or may not choose to disambiguate.

Senses of *plant* in Wordnet

- plant-1: works, industrial plant (buildings for carrying on industrial labor) "they built a large plant to manufacture automobiles"
- plant-2: flora, plant life ((botany) a living organism lacking the power of locomotion)

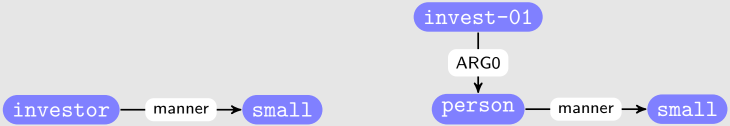
Senses of *keep* in Propbank

- keep.01(ARG0:Keeper, ARG1:thing-kept): "The Herald kept its old-time Hearst readership."
- keep.02(ARG0:causer-of-continued-action, ARG1:continued-entity, ARG2:continued-state-or-action): "The captain kept the crew loyal."

Lexical Decomposition

The meanings of individual words can have internal structure, which the semantic annotation may or may not represent.

How to represent "small investor"?



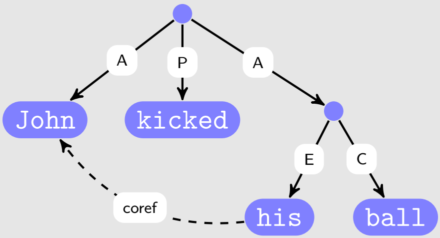
Anaphoric Coreference

The meaning of an anaphoric expression depends on the context in which it occurs (within the sentence; across sentences).

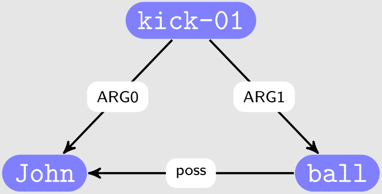
Examples

- John kicked his ball.
- He wants her to see him.

Modeling choices



UCCA



AMR

Summary: Facets of Linguistic Meaning

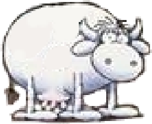
- Predicate-argument structure
- Quantification and Scope
- Presupposition and focus
- Word sense differentiation
- Lexical decomposition
- Anaphoric coreference

Summary: Facets of Linguistic Meaning

- Predicate-argument structure
- Quantification and Scope
- Presupposition and focus
- Word sense differentiation
- Lexical decomposition
- Anaphoric coreference

- Grounding (in world; in picture; in Wikipedia; . . .)
- Tense and aspect
- Information structure
- Discourse structure
- . . . and many others . . .

dotted + **cow** = **dotted cow**



(Example by Jan van Eijck)

The Principle of Compositionality

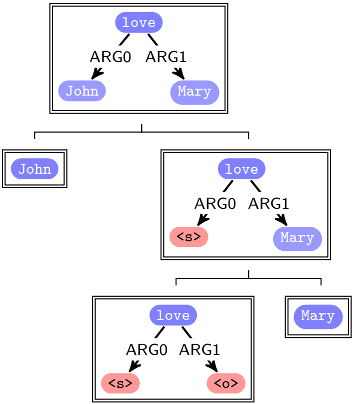
The meaning of an expression is a function of the meanings of its parts and of the way they are syntactically combined. *B. Partee*

Compositionality

Not all semantic phenomena lend themselves easily to a compositional analysis.

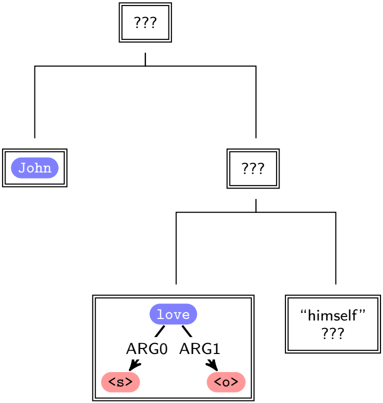
Predicate–argument structure

"John loves Mary."



Coreference

"John loves himself."



2 Foundations: Basic Graph Theory

$$\mathbb{G} = \langle N, E, T \rangle$$

- $\mathbb{G}$ is a directed graph: N is set of nodes; $E \subseteq N \times N$ is set of edges;
- $T \subseteq N$ is possibly empty set of top node(s): the 'main' predicate(s);

$$\mathbb{G} = \langle N, E, T \rangle$$

- $\mathbb{G}$ is a directed graph: N is set of nodes; $E \subseteq N \times N$ is set of edges;
- $T \subseteq N$ is possibly empty set of top node(s): the 'main' predicate(s);
- ***in***- and ***out***-degree of $n \in N$ count edges to and from n; ***in*** $= 0$: root;
- top in *Abrams arrived quickly.* is *arrive*, but can be argument of *quick*;
- semantic graphs often multi-rooted: rootness just a structural property;

$$\mathbb{G} = \langle N, E, T \rangle$$

- $\mathbb{G}$ is a directed graph: N is set of nodes; $E \subseteq N \times N$ is set of edges;
- $T \subseteq N$ is possibly empty set of top node(s): the 'main' predicate(s);
- ***in***- and ***out***-degree of $n \in N$ count edges to and from n; ***in*** $= 0$: root;
- top in *Abrams arrived quickly.* is *arrive*, but can be argument of *quick*;
- semantic graphs often multi-rooted: rootness just a structural property;
- a node n is reentrant if ***in***$(n) > 1$ (shared argument across predicates);

$$\mathbb{G} = \langle N, E, T \rangle$$

- $\mathbb{G}$ is a directed graph: N is set of nodes; $E \subseteq N \times N$ is set of edges;
- $T \subseteq N$ is possibly empty set of top node(s): the 'main' predicate(s);
- ***in***- and ***out***-degree of $n \in N$ count edges to and from n; ***in*** $= 0$: root;
- top in *Abrams arrived quickly.* is *arrive*, but can be argument of *quick*;
- semantic graphs often multi-rooted: rootness just a structural property;
- a node n is reentrant if ***in***$(n) > 1$ (shared argument across predicates);
- cycles can occur: directed path from m to n and ('back') from n to m;

$$\mathbb{G} = \langle N, E, T \rangle$$

- $\mathbb{G}$ is a directed graph: N is set of nodes; $E \subseteq N \times N$ is set of edges;
- $T \subseteq N$ is possibly empty set of top node(s): the 'main' predicate(s);
- ***in***- and ***out***-degree of $n \in N$ count edges to and from n; ***in*** $= 0$: root;
- top in *Abrams arrived quickly.* is *arrive*, but can be argument of *quick*;
- semantic graphs often multi-rooted: rootness just a structural property;
- a node n is reentrant if ***in***$(n) > 1$ (shared argument across predicates);
- cycles can occur: directed path from m to n and ('back') from n to m;
- $\mathbb{G}$ is connected if there is an undirected path between all pairs of nodes;

$$\mathbb{G} = \langle N, E, T \rangle$$

- $\mathbb{G}$ is a directed graph: N is set of nodes; $E \subseteq N \times N$ is set of edges;
- $T \subseteq N$ is possibly empty set of top node(s): the 'main' predicate(s);
- ***in***- and ***out***-degree of $n \in N$ count edges to and from n; ***in*** $= 0$: root;
- top in *Abrams arrived quickly.* is *arrive*, but can be argument of *quick*;
- semantic graphs often multi-rooted: rootness just a structural property;
- a node n is reentrant if ***in***$(n) > 1$ (shared argument across predicates);
- cycles can occur: directed path from m to n and ('back') from n to m;
- $\mathbb{G}$ is connected if there is an undirected path between all pairs of nodes;
- $\mathbb{G}$ is a tree if $|T| = 1$ and there is a unique path to all other nodes.

Graph Structure vs. Node and Edge Decorations

Predicates vs. Constants

- Nodes and edges can be labeled (e.g. by relation and role identifiers);

Graph Structure vs. Node and Edge Decorations

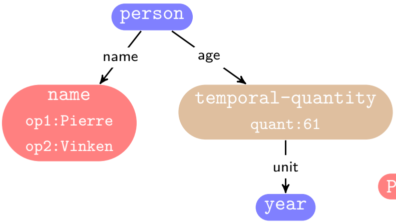
Predicates vs. Constants

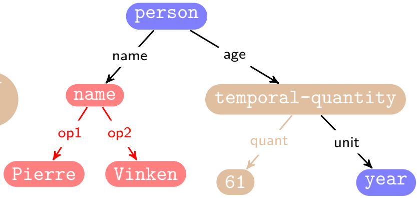
- Nodes and edges can be labeled (e.g. by relation and role identifiers);
- labels can be internally structured: node properties and edge attributes;
- properties (and attributes) are non-recursive attribute–value matrices;
- node (and edge) label is merely a distinguished property (or attribute);

Graph Structure vs. Node and Edge Decorations

Predicates vs. Constants

- Nodes and edges can be labeled (e.g. by relation and role identifiers);
- labels can be internally structured: node properties and edge attributes;
- properties (and attributes) are non-recursive attribute–value matrices;
- node (and edge) label is merely a distinguished property (or attribute);
- distinction is not commonly discussed, but used by many frameworks.





Anchoring in the Surface String

Relating Pieces of Meaning to the Linguistic Signal

- Intuitively, sub-structures of meaning relate to sub-parts of the signal;
- semantic frameworks vary in how much weight to put on these relations;

Anchoring in the Surface String

Relating Pieces of Meaning to the Linguistic Signal

- Intuitively, sub-structures of meaning relate to sub-parts of the signal;
- semantic frameworks vary in how much weight to put on these relations;
- anchoring of graph elements in sub-strings of the underlying utterance;
- can be part of semantic annotations or not; can take different forms;

Anchoring in the Surface String

Relating Pieces of Meaning to the Linguistic Signal

- Intuitively, sub-structures of meaning relate to sub-parts of the signal;
- semantic frameworks vary in how much weight to put on these relations;
- anchoring of graph elements in sub-strings of the underlying utterance;
- can be part of semantic annotations or not; can take different forms;
- hierarchy of anchoring types: Flavor (0)–(2); bilexical graphs strictest;

Flavor	Name	Example	Type of Anchoring
0	bilexical	SDP	nodes are sub-set of surface tokens
1	anchored	EDS	free node–sub-string correspondences
2	unanchored	AMR	no sub-string correspondences annotated

Anchoring in the Surface String

Relating Pieces of Meaning to the Linguistic Signal

- Intuitively, sub-structures of meaning relate to sub-parts of the signal;
- semantic frameworks vary in how much weight to put on these relations;
- anchoring of graph elements in sub-strings of the underlying utterance;
- can be part of semantic annotations or not; can take different forms;
- hierarchy of anchoring types: Flavor (0)–(2); bilexical graphs strictest;
- anchoring central in parsing, explicit or latent; aka ‘alignment’ for AMR;
- relevant to at least some downstream tasks; can impact evaluation.

Flavor	Name	Example	Type of Anchoring
0	bilexical	SDP	nodes are sub-set of surface tokens
1	anchored	EDS	free node–sub-string correspondences
2	unanchored	AMR	no sub-string correspondences annotated

3 Semantic Graphbanks

Overview: A Selection of Graphbanks

Selection Criteria

- ‘Full-sentence’ semantics: all content-bearing units receive annotations;
- natively graph-based: meaning representation through (directed) graphs;
- large-scale, gold-standard annotations and parsers are publicly available.

Overview: A Selection of Graphbanks

Selection Criteria

- 'Full-sentence' semantics: all content-bearing units receive annotations;
- natively graph-based: meaning representation through (directed) graphs;
- large-scale, gold-standard annotations and parsers are publicly available.

Semantic Graphbanks

- Bilexical semantic dependencies: DM, PAS, PSD, CCD;
- Variants of English Resource Semantics: EDS, (DMRS, DM);
- Abstract Meaning Representations: AMR;
- Universal Conceptual Cognitive Annotation: UCCA.

Overview: A Selection of Graphbanks

Selection Criteria

- 'Full-sentence' semantics: all content-bearing units receive annotations;
- natively graph-based: meaning representation through (directed) graphs;
- large-scale, gold-standard annotations and parsers are publicly available.

Semantic Graphbanks

- Bilexical semantic dependencies: DM, PAS, PSD, CCD;
- Variants of English Resource Semantics: EDS, (DMRS, DM);
- Abstract Meaning Representations: AMR;
- Universal Conceptual Cognitive Annotation: UCCA.

(With Apologies to) Non-Graph or Non-Meaning Banks

- PropBank (Palmer et al., 2005), Framenet (Baker et al., 1998), . . . ;
- Groningen Parallel Meaning Bank: GMB, PMB (Basile et al., 2012);
- Universal Decompositional Semantics (White et al., 2016);
- Enhanced Universal Dependencies (Schuster & Manning, 2016);
- . . .

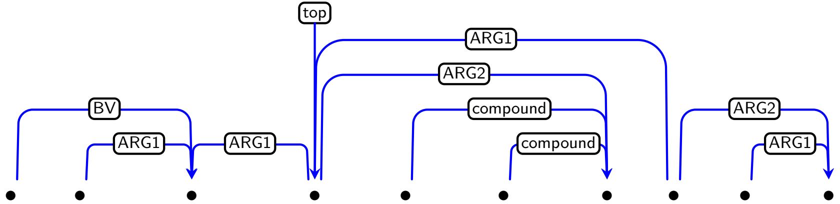
Arguably Basicest: Bilexical Semantic Dependencies

- Two decades of great advances in syntactic dependencies and parsing;
- recently, renewed interest in meaning; algorithmic interest in graphs;

Arguably Basicest: Bilexical Semantic Dependencies

- Two decades of great advances in syntactic dependencies and parsing;
- recently, renewed interest in meaning; algorithmic interest in graphs;
- nodes limited to surface lexical units (words):

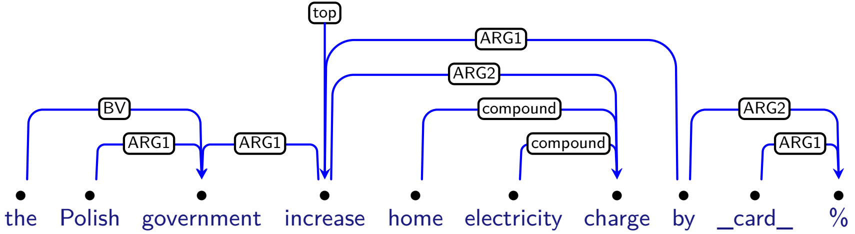
The Polish government increased home electricity charges by 150%. [WSJ #0037059]



Arguably Basicest: Bilexical Semantic Dependencies

- Two decades of great advances in syntactic dependencies and parsing;
- recently, renewed interest in meaning; algorithmic interest in graphs;
- nodes limited to surface lexical units (words): lemmas

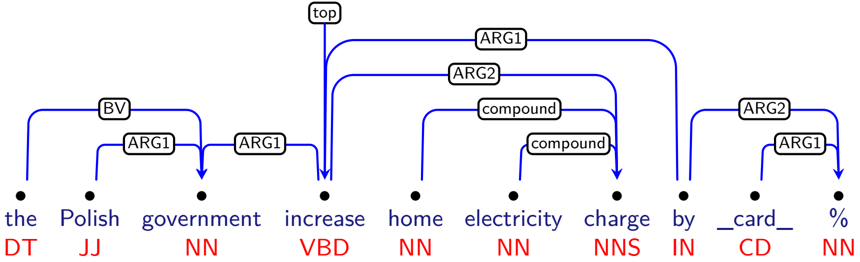
The Polish government increased home electricity charges by 150%. [WSJ #0037059]



Arguably Basicest: Bilexical Semantic Dependencies

- Two decades of great advances in syntactic dependencies and parsing;
- recently, renewed interest in meaning; algorithmic interest in graphs;
- nodes limited to surface lexical units (words): lemmas, PoS

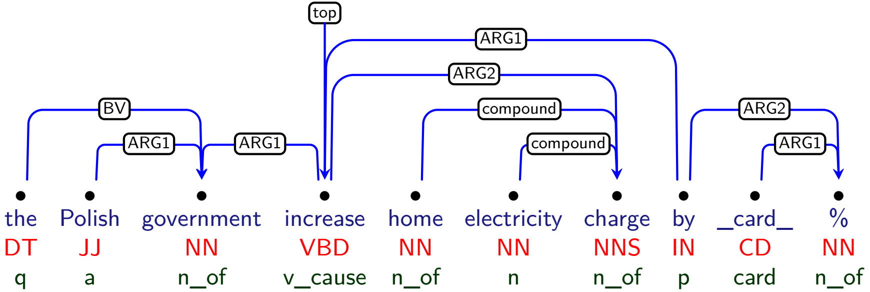
The Polish government increased home electricity charges by 150%. [WSJ #0037059]



Arguably Basicest: Bilexical Semantic Dependencies

- Two decades of great advances in syntactic dependencies and parsing;
- recently, renewed interest in meaning; algorithmic interest in graphs;
- nodes limited to surface lexical units (words): lemmas, PoS, and frames;

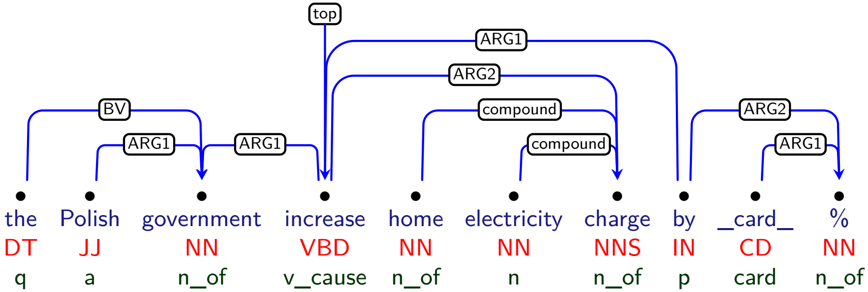
The Polish government increased home electricity charges by 150%. [WSJ #0037059]



Arguably Basicest: Bilexical Semantic Dependencies

- Two decades of great advances in syntactic dependencies and parsing;
- recently, renewed interest in meaning; algorithmic interest in graphs;
- nodes limited to surface lexical units (words): lemmas, PoS, and frames;
- edges encode argument roles and maybe some construction semantics;

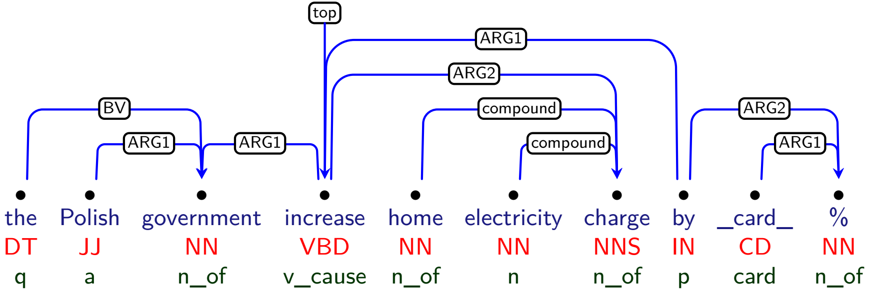
The Polish government increased home electricity charges by 150%. [WSJ #0037059]



Arguably Basicest: Bilexical Semantic Dependencies

- Two decades of great advances in syntactic dependencies and parsing;
- recently, renewed interest in meaning; algorithmic interest in graphs;
- nodes limited to surface lexical units (words): lemmas, PoS, and frames;
- edges encode argument roles and maybe some construction semantics;
- limited expressivity, e.g. no lexical decomposition, no covert meaning.

The Polish government increased home electricity charges by 150%. [WSJ #0037059]



The Semantic Dependency Parsing Tasks (SDP)

Background

- Parallel work on bilexical semantic dependencies over the same corpus;
- different linguistic traditions, syntactico-semantic frameworks, schools.

The Semantic Dependency Parsing Tasks (SDP)

Background

- Parallel work on bilexical semantic dependencies over the same corpus;
- different linguistic traditions, syntactico-semantic frameworks, schools.

SemEval 2014 (Oepen et al., 2014)

- Sentence- and token-align annotations; simplify into bilexical digraphs;
- 34,004 + 1348 sentences, 745,543 + 29,808 tokens from WSJ corpus;
- three frameworks: DM, PAS, PSD; nine participating teams: 78–92 F_1.

The Semantic Dependency Parsing Tasks (SDP)

Background

- Parallel work on bilexical semantic dependencies over the same corpus;
- different linguistic traditions, syntactico-semantic frameworks, schools.

SemEval 2014 (Oepen et al., 2014)

- Sentence- and token-align annotations; simplify into bilexical digraphs;
- 34,004 + 1348 sentences, 745,543 + 29,808 tokens from WSJ corpus;
- three frameworks: DM, PAS, PSD; nine participating teams: 78–92 F_1.

SemEval 2015 (Oepen et al., 2015)

+ Out-of-domain English test data from Brown Corpus: 1,849; 31,583;
+ Chinese PAS (32,783; 687,433) and Czech PSD (48,972; 1,111,626);
+ frame (or sense) prediction; evaluation beyond ‘atomic’ dependency F_1.

The Semantic Dependency Parsing Tasks (SDP)

Background

- Parallel work on bilexical semantic dependencies over the same corpus;
- different linguistic traditions, syntactico-semantic frameworks, schools.

SemEval 2014 (Oepen et al., 2014)

- Sentence- and token-align annotations; simplify into bilexical digraphs;
- 34,004 + 1348 sentences, 745,543 + 29,808 tokens from WSJ corpus;
- three frameworks: DM, PAS, PSD; nine participating teams: 78–92 F_1.

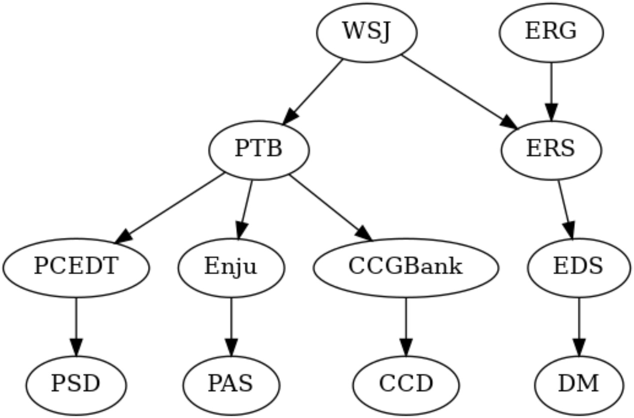
SemEval 2015 (Oepen et al., 2015)

+ Out-of-domain English test data from Brown Corpus: 1,849; 31,583;
+ Chinese PAS (32,783; 687,433) and Czech PSD (48,972; 1,111,626);
+ frame (or sense) prediction; evaluation beyond 'atomic' dependency F_1.

More Recently (`http://sdp.delph-in.net`)

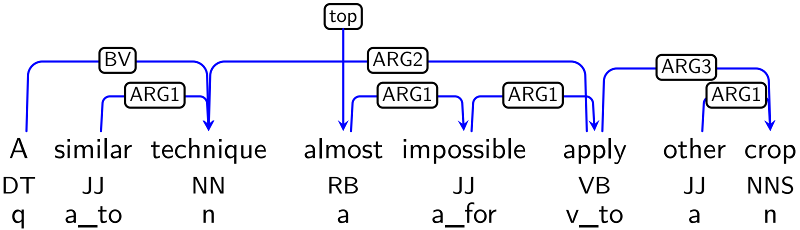
- Release as LDC2016T10 with fourth framework; 'standard' benchmark.

Bilexical Semantic Dependencies: English Genealogy



DELPH-IN MRS Bi-Lexical Dependencies (DM)

A similar technique is almost impossible to apply to other crops.

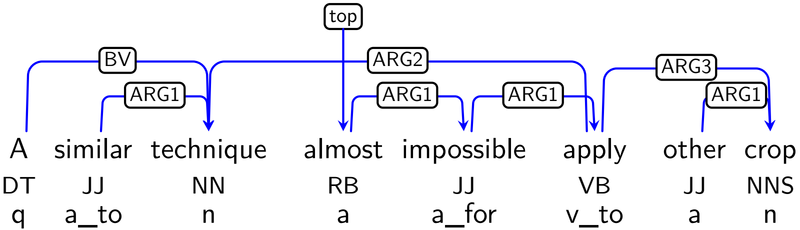


Ivanova et al. (2012)

- Simplification from underspecified logical forms (ERS; coming right up);
- edge labels are mostly semantic argument positions, e.g. *apply'*$(\_, \_, \_)$;

DELPH-IN MRS Bi-Lexical Dependencies (DM)

A similar technique is almost impossible to apply to other crops.

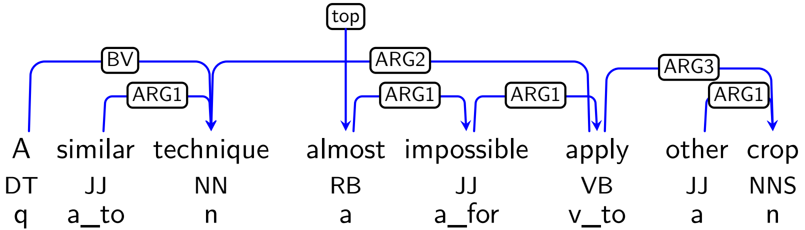


Ivanova et al. (2012)

- Simplification from underspecified logical forms (ERS; coming right up);
- edge labels are mostly semantic argument positions, e.g. *apply'*(_, _, _);
- PoS and frames for coarse sense differentiation (*apply* ... *to* vs. ... *for*);

DELPH-IN MRS Bi-Lexical Dependencies (DM)

A similar technique is almost impossible to apply to other crops.



Ivanova et al. (2012)

- Simplification from underspecified logical forms (ERS; coming right up);
- edge labels are mostly semantic argument positions, e.g. *apply'*$(\_, \_, \_)$;
- PoS and frames for coarse sense differentiation (*apply ... to* vs. *... for*);
- typically, unique top node; quantifiers have special 'bound variable' role.

Background: English Resource Semantics (ERS)

LinGO English Resource Grammar (Flickinger, 2000; Flickinger et al., 2017)

- Hand-designed computational grammar for English in HPSG framework;
- declarative, unification-based: parsing and realization; multiple engines;
- 25^{+} person years; coverage of 85–95 % of running text across domains;

Background: English Resource Semantics (ERS)

LinGO English Resource Grammar (Flickinger, 2000; Flickinger et al., 2017)

- Hand-designed computational grammar for English in HPSG framework;
- declarative, unification-based: parsing and realization; multiple engines;
- 25^{+} person years; coverage of 85–95 % of running text across domains;
- underspecified meaning representation in MRS (Copestake et al., 2005).

Background: English Resource Semantics (ERS)

LinGO English Resource Grammar (Flickinger, 2000; Flickinger et al., 2017)

- Hand-designed computational grammar for English in HPSG framework;
- declarative, unification-based: parsing and realization; multiple engines;
- 25^{+} person years; coverage of 85–95 % of running text across domains;
- underspecified meaning representation in MRS (Copestake et al., 2005).

LinGO Redwoods Treebank (Carter, 1997; Oepen et al., 2004)

- Grammar-based annotation: select 'correct' reading from parse forest;
- efficient selection based on minimal discriminants; re-use across releases;

Background: English Resource Semantics (ERS)

LinGO English Resource Grammar (Flickinger, 2000; Flickinger et al., 2017)

- Hand-designed computational grammar for English in HPSG framework;
- declarative, unification-based: parsing and realization; multiple engines;
- 25^{+} person years; coverage of 85–95 % of running text across domains;
- underspecified meaning representation in MRS (Copestake et al., 2005).

LinGO Redwoods Treebank (Carter, 1997; Oepen et al., 2004)

- Grammar-based annotation: select 'correct' reading from parse forest;
- efficient selection based on minimal discriminants; re-use across releases;
- version 1214: some 85,000 annotated sentences, six^{+} different domains;
- including Sections 00–21 from the venerable WSJ Corpus; sub-set of Brown Corpus; Wikipedia; tourism; ecommerce; transcribed speech;

Background: English Resource Semantics (ERS)

LinGO English Resource Grammar (Flickinger, 2000; Flickinger et al., 2017)

- Hand-designed computational grammar for English in HPSG framework;
- declarative, unification-based: parsing and realization; multiple engines;
- 25^+ person years; coverage of 85–95 % of running text across domains;
- underspecified meaning representation in MRS (Copestake et al., 2005).

LinGO Redwoods Treebank (Carter, 1997; Oepen et al., 2004)

- Grammar-based annotation: select 'correct' reading from parse forest;
- efficient selection based on minimal discriminants; re-use across releases;
- version 1214: some 85,000 annotated sentences, six^+ different domains;
- including Sections 00–21 from the venerable WSJ Corpus; sub-set of Brown Corpus; Wikipedia; tourism; ecommerce; transcribed speech;
- Bender et al. (2015) report inter-annotator agreement of 0.94 EDM_{na};

Background: English Resource Semantics (ERS)

LinGO English Resource Grammar (Flickinger, 2000; Flickinger et al., 2017)

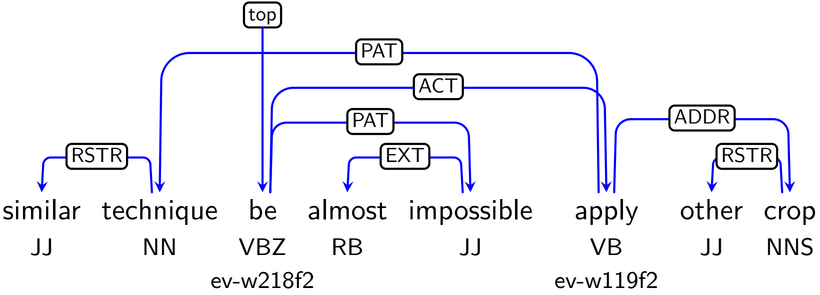
- Hand-designed computational grammar for English in HPSG framework;
- declarative, unification-based: parsing and realization; multiple engines;
- 25^{+} person years; coverage of 85–95 % of running text across domains;
- underspecified meaning representation in MRS (Copestake et al., 2005).

LinGO Redwoods Treebank (Carter, 1997; Oepen et al., 2004)

- Grammar-based annotation: select 'correct' reading from parse forest;
- efficient selection based on minimal discriminants; re-use across releases;
- version 1214: some 85,000 annotated sentences, six^{+} different domains;
- including Sections 00–21 from the venerable WSJ Corpus; sub-set of Brown Corpus; Wikipedia; tourism; ecommerce; transcribed speech;
- Bender et al. (2015) report inter-annotator agreement of 0.94 EDM_{na};
- record full HPSG derivation; export into various graph-based formats.

Prague Semantic Dependencies (PSD)

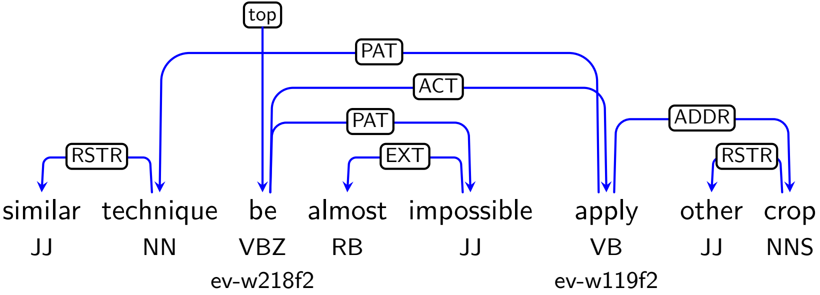
A similar technique is almost impossible to apply to other crops.



- Simplification of multi-layer dependency annotation (coming right up);

Prague Semantic Dependencies (PSD)

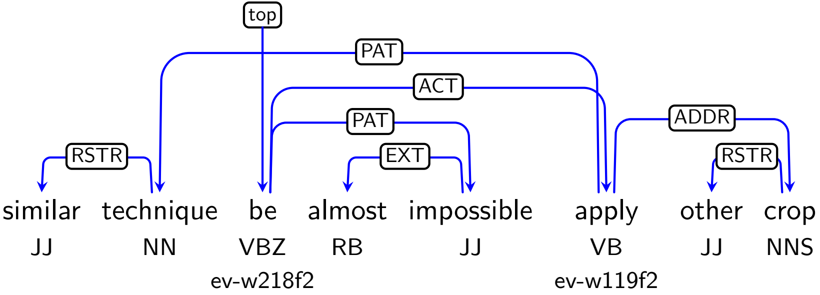
A similar technique is almost impossible to apply to other crops.



- Simplification of multi-layer dependency annotation (coming right up);
- ACT(or), PAT(ient), ADDR(essee), ORIG(in), EFF(ect): complements;
- first verbal argument is ACT, also in causative–inchoative alternation;

Prague Semantic Dependencies (PSD)

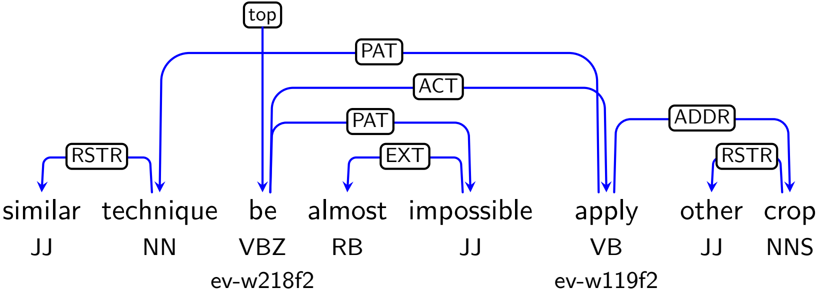
A similar technique is almost impossible to apply to other crops.



- Simplification of multi-layer dependency annotation (coming right up);
- ACT(or), PAT(ient), ADDR(essee), ORIG(in), EFF(ect): complements;
- first verbal argument is ACT, also in causative–inchoative alternation;
- unlike DM, adjuncts are dependents, e.g. EXT(end) or RSTR(iction);

Prague Semantic Dependencies (PSD)

A similar technique is almost impossible to apply to other crops.



- Simplification of multi-layer dependency annotation (coming right up);
- ACT(or), PAT(ient), ADDR(essee), ORIG(in), EFF(ect): complements;
- first verbal argument is ACT, also in causative–inchoative alternation;
- unlike DM, adjuncts are dependents, e.g. EXT(end) or RSTR(iction);
- simple determiners (*a*, *the*) treated as not content-bearning (in PSD).

Background: Functional Generative Description

Linguistic Theory (Sgall et al., 1986)

- Charles University in Prague, since 1960s; (Petr Sgall; 1926–2019);
- layered dependencies, e.g. morphological, analytical, tectogrammatical;
- nodes are interlinked across layers; simple feature structures on nodes;
- tectogrammatical layer: predicate–argument structure, verb senses, coreference, ellipsis, information structure, discourse connectives, . . .
- formally, always rooted trees; ‘generated’ nodes and explicit coreference.

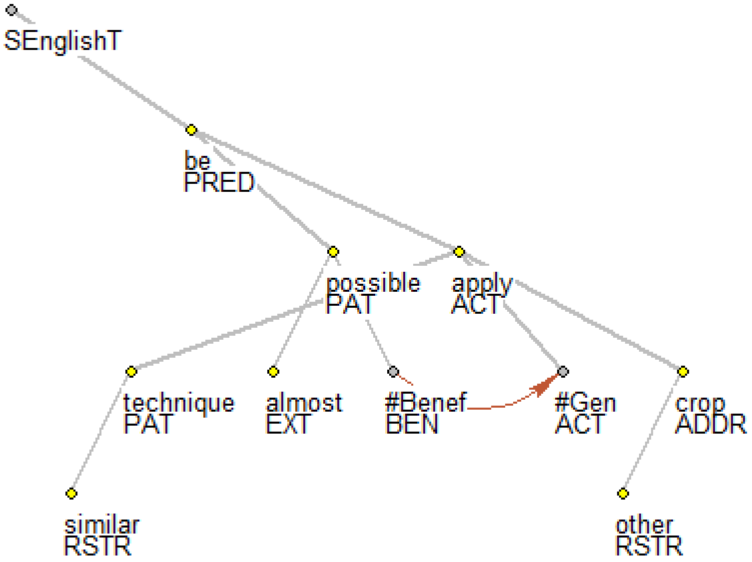
Background: Functional Generative Description

Linguistic Theory (Sgall et al., 1986)

- Charles University in Prague, since 1960s; (Petr Sgall; 1926–2019);
- layered dependencies, e.g. morphological, analytical, tectogrammatical;
- nodes are interlinked across layers; simple feature structures on nodes;
- tectogrammatical layer: predicate–argument structure, verb senses, coreference, ellipsis, information structure, discourse connectives, ...
- formally, always rooted trees; 'generated' nodes and explicit coreference.

Large-Scale Annotations (Böhmová et al., 2003; Hajič et al., 2012)

- Prague Dependency Treebank (PDT) 3.5: 800,000 tokens of Czech;
- Prague Czech–English Dependency Treebank (PCEDT) 2.0: PTB WSJ;
- since mid-1990s, several person decades of annotation; still on-going;
- pre-SDP, some use in morpho-syntactic parsing and machine translation.



http://ufal.mff.cuni.cz/pedt2.0/introduction.html

Background: Tectogrammatical Valency Frames

apply[2] **ACT**(sub) **PAT**(obj1;ving) **ADDR**(to[objpp;ving])

- *Mr. Bologna said the sale would allow Gen-Probe to speed up the development of new technology, and to more quickly apply existing technology to an array of diagnostic products.*
- *The gains are to be applied *trace* to fourth quarter or first-quarter results.*

Corpus example(s): Close [X]

pedt Saudi **Arabia**←ACT, for its part, has vowed *-2 to enact a copyright law compatible with international standards and *-2 **to apply the** PAT **law** PAT **to** ADDR computer **software** ADDR as well as **to** ADDR literary **works** ADDR , Mrs. Hills said *T*-1.

pedt **It** PAT **was** later **applied to** ADDR other new-car **programs** ADDR **,** ADDR including those that *T*-1 produced the Ford Thunderbird and Mercury Cougar**.**

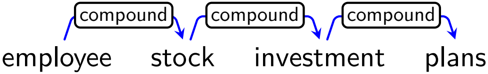
pedt **A** PAT similar **technique** PAT is almost impossible 0 * **to apply** *T*-1 **to** ADDR other **crops** ADDR , such as **cotton** ADDR , **soybeans** ADDR and **rice** ADDR .

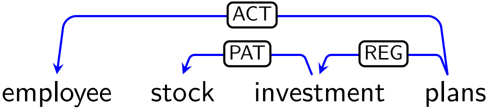
pedt Armstrong expects *-1 to close the sale of the color unit in late November and the carpet sale in December, **with the** PAT **gains** PAT **to be applied** *-4 **to** ADDR fourth quarter or first-quarter **results** ADDR .

pedt *-1 Using small electrical **shocks**←PAT **applied** * **to** ADDR her **feet** ADDR , they were able *-1 to monitor sensory nerves.

http://lindat.mff.cuni.cz/services/EngVallex/EngVallex.html

DM vs. PSD: Sentence vs. Speaker Meaning?

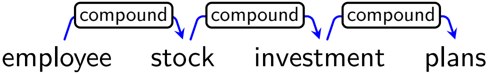


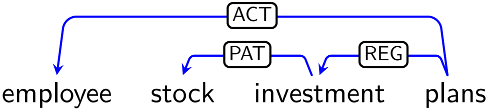


Diverging Ambitions

- Meaning determined by linguistic signal alone vs. by utterance context;
- internal bracketing arguably grammaticized, but not role interpretation;

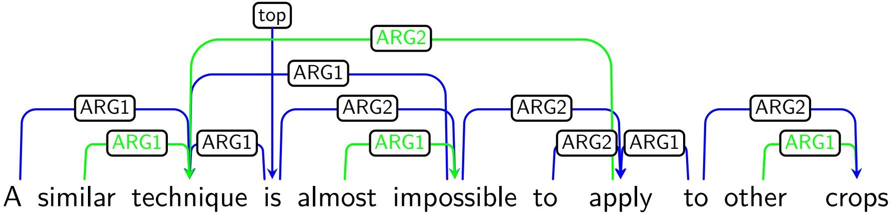
DM vs. PSD: Sentence vs. Speaker Meaning?





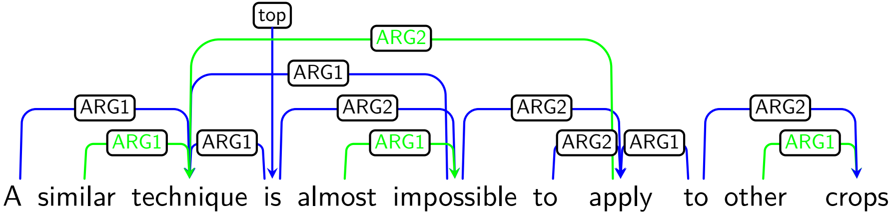
Diverging Ambitions

- Meaning determined by linguistic signal alone vs. by utterance context;
- internal bracketing arguably grammaticized, but not role interpretation;
- *investment* and *plan* as deverbal: PSD valency frames parallel to verbs;
- never mind disagreement in bracketing (both structures are defensible).



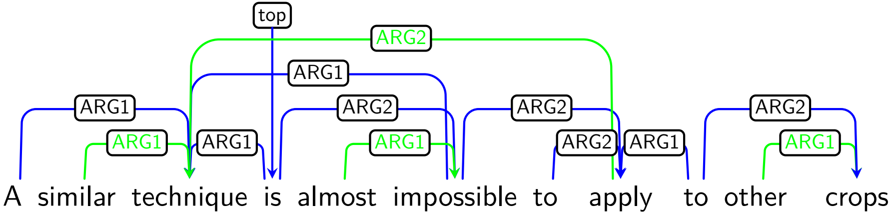
PAS: Enju Predicate–Argument Structures (Miyao, 2006)

- Similar in pedigree to DM: derived compositionally by lage-scale HPSG;
- Enju Treebank: Mostly automatic conversion from PTB; limited syntax;



PAS: Enju Predicate–Argument Structures (Miyao, 2006)

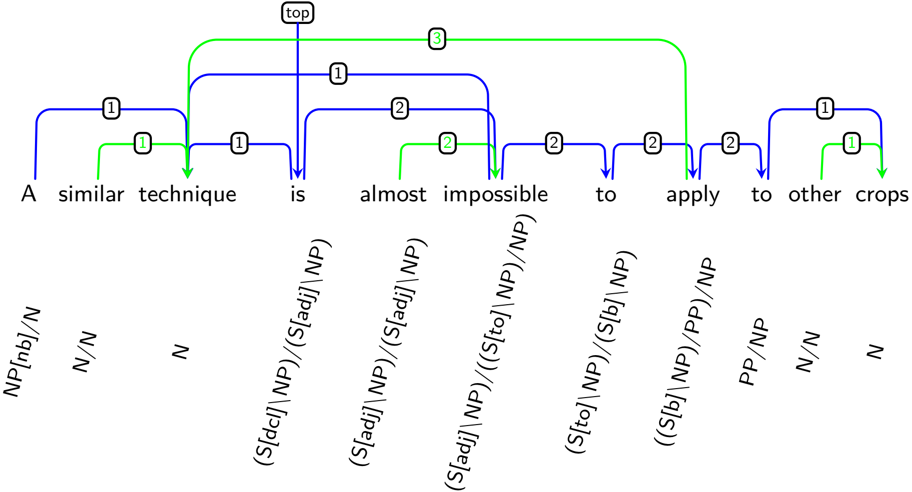
- Similar in pedigree to DM: derived compositionally by lage-scale HPSG;
- Enju Treebank: Mostly automatic conversion from PTB; limited syntax;
- missing lexical knowledge, e.g. ARG3 (oblique complement) of *apply*;
- like in CCD, several syntactic dependencies, e.g. *technique* as ‘subject’;



PAS: Enju Predicate–Argument Structures (Miyao, 2006)

- Similar in pedigree to DM: derived compositionally by lage-scale HPSG;
- Enju Treebank: Mostly automatic conversion from PTB; limited syntax;
- missing lexical knowledge, e.g. ARG3 (oblique complement) of *apply*;
- like in CCD, several syntactic dependencies, e.g. *technique* as 'subject';
- Enju Parser was early broad-coverage engine for semantic dependencies.

Other Bilexical ‘Semantic’ Dependencies (2 of 2)



CCD: CCG Word–Word Dependencies (Hockenmaier & Steedman, 2007)

- CCG categories as ‘frame’ identifiers; edge labels for argument position;
- more ‘deep syntax’ than semantics, but functor–argument directionality.

Pairwise Similarity (Unlabeled Dependency F_1)

	Directed			Undirected		
	DM	PAS	PSD	DM	PAS	PSD
DM	–	.6425	.2612	–	.6719	.5675
PAS	.6688	–	.2963	.6993	–	.5490
PSD	.2636	.2963	–	.5743	.5630	–

(Upper Right Diagonals: Including punctuation; Lower Left: Ignoring It)

Pairwise Similarity (Unlabeled Dependency F_1)

	Directed			Undirected		
	DM	PAS	PSD	DM	PAS	PSD
DM	–	.6425	.2612	–	.6719	.5675
PAS	.6688	–	.2963	.6993	–	.5490
PSD	.2636	.2963	–	.5743	.5630	–

(Upper Right Diagonals: Including punctuation; Lower Left: Ignoring It)

- DM and PAS structurally much closer to each other than either to PSD;
- effect stronger when ignoring dependencies involving punctuation marks;

Pairwise Similarity (Unlabeled Dependency F_1)

	Directed			Undirected		
	DM	PAS	PSD	DM	PAS	PSD
DM	–	.6425	.2612	–	.6719	.5675
PAS	.6688	–	.2963	.6993	–	.5490
PSD	.2636	.2963	–	.5743	.5630	–

(Upper Right Diagonals: Including punctuation; Lower Left: Ignoring It)

- DM and PAS structurally much closer to each other than either to PSD;
- effect stronger when ignoring dependencies involving punctuation marks;
- directionality of dependencies as one of the key sources of divergence.

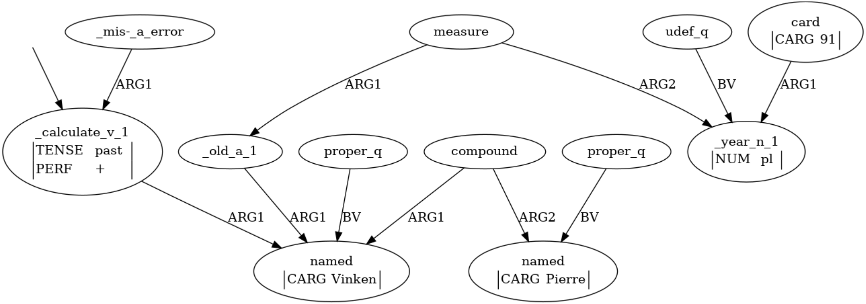
Moving On: Elementary Dependency Structures (EDS)

Limitations in Bi-Lexical Semantic Dependencies

- Challenges: lexical decomposition, sub-lexical or construction semantics.

Limitations in Bi-Lexical Semantic Dependencies

- Challenges: lexical decomposition, sub-lexical or construction semantics.

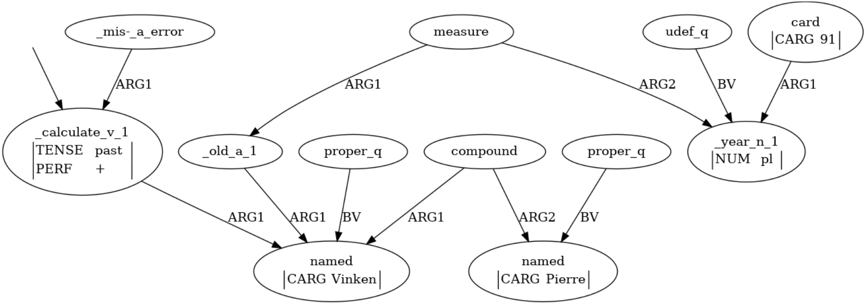


Pierre Vinken, 91 years old, had mis-calculated.

Moving On: Elementary Dependency Structures (EDS)

Limitations in Bi-Lexical Semantic Dependencies

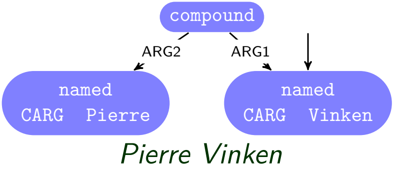
- Challenges: lexical decomposition, sub-lexical or construction semantics.



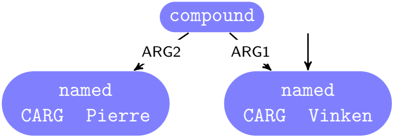
Pierre Vinken, 91 years old, had mis-calculated.

- ERS without variables (Oepen et al., 2002; Oepen & Lønning, 2006);
- like DM, nodes encode n-ary predications, edges (expressed) arguments;
- Flavor (1): arbitrary (overlapping) sub-strings carry pieces of meaning.

EDSs are ‘Radically Compositional’



EDSs are 'Radically Compositional'



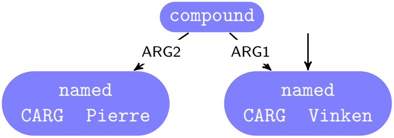
Pierre Vinken

Named Entities

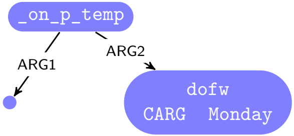
- Underspecified structure in names;
- few, lexically determined sub-types.

Michelle and Barack Obama

EDSs are 'Radically Compositional'



Pierre Vinken



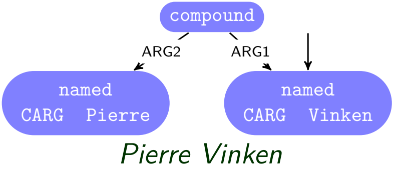
on Monday

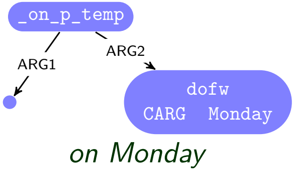
Named Entities

- Underspecified structure in names;
- few, lexically determined sub-types.

Michelle and Barack Obama

EDSs are 'Radically Compositional'





Named Entities

- Underspecified structure in names;
- few, lexically determined sub-types.

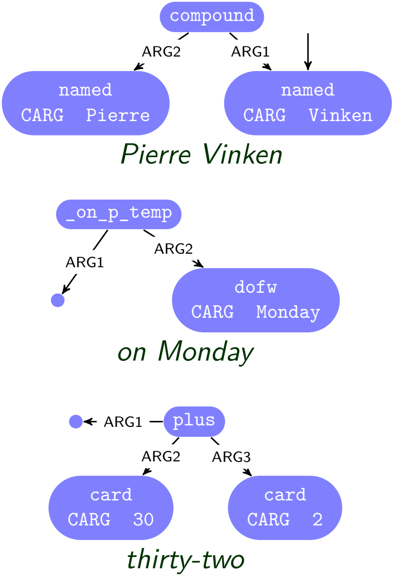
Michelle and Barack Obama

Prepositions (and Similar)

- Predicates: distinct two-place relation;
- specialized sub-senses as appropriate.

before and during the meeting

EDSs are 'Radically Compositional'



Named Entities

- Underspecified structure in names;
- few, lexically determined sub-types.

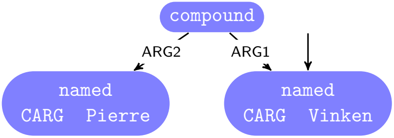
Michelle and Barack Obama

Prepositions (and Similar)

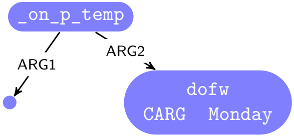
- Predicates: distinct two-place relation;
- specialized sub-senses as appropriate.

before and during the meeting

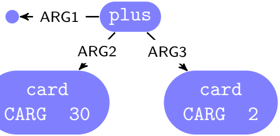
EDSs are 'Radically Compositional'



Pierre Vinken



on Monday



thirty-two

Named Entities

- Underspecified structure in names;
- few, lexically determined sub-types.

Michelle and Barack Obama

Prepositions (and Similar)

- Predicates: distinct two-place relation;
- specialized sub-senses as appropriate.

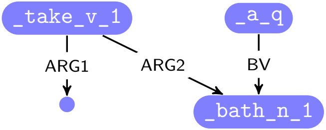
before and during the meeting

Literal Numbers

- syntax yields arithmetic expressions;
- trivial 'downstream' normalization.

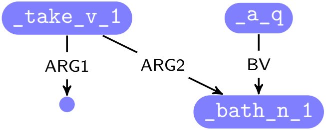
ten two twenty thousand

More Examples of Compositionality in EDS



take a bath

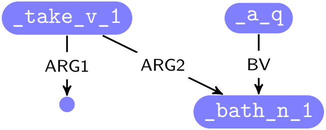
More Examples of Compositionality in EDS



take a bath

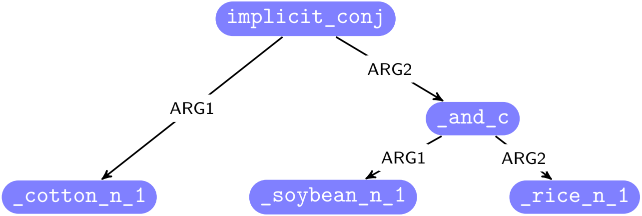
She took a soothing bath.

More Examples of Compositionality in EDS



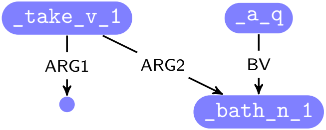
take a bath

She took a soothing bath.



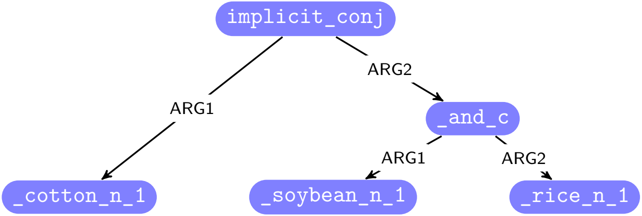
cotton, soybeans, and rice

More Examples of Compositionality in EDS



take a bath

She took a soothing bath.



cotton, soybeans, and rice

She arrived, ate, and slept.

A Few More Reflections on EDS

Sense Differentiation

- Focus on sentence meaning: dinstinguish grammaticalized contrasts:
- *look up the answer* (`_look_v_up`) vs. *look up the hill* (`_look_v_1`);

A Few More Reflections on EDS

Sense Differentiation

- Focus on sentence meaning: dinstinguish grammaticalized contrasts:
- *look up the answer* (`_look_v_up`) vs. *look up the hill* (`_look_v_1`);
- *he broke the vase* (`_break_v_cause`) vs. *the vase broke* (`_break_v_1`);

A Few More Reflections on EDS

Sense Differentiation

- Focus on sentence meaning: dinstinguish grammaticalized contrasts:
- *look up the answer* (`_look_v_up`) vs. *look up the hill* (`_look_v_1`);
- *he broke the vase* (`_break_v_cause`) vs. *the vase broke* (`_break_v_1`);
- no grammatical contrast in *draw a house* vs. *draw a cart* (`_draw_v_1`).

A Few More Reflections on EDS

Sense Differentiation

- Focus on sentence meaning: dinstinguish grammaticalized contrasts:
- *look up the answer* (`_look_v_up`) vs. *look up the hill* (`_look_v_1`);
- *he broke the vase* (`_break_v_cause`) vs. *the vase broke* (`_break_v_1`);
- no grammatical contrast in *draw a house* vs. *draw a cart* (`_draw_v_1`).

Limitations

- Grammar-based annotation: no ‘correct’ analysis in 5–15 % of inputs;
- limited resources for other languages (German, Japanese, Spanish, . . .);
- tense, aspect, mood, number, etc. as ‘morphological’ node properties;
- partial information about scope discarded in conversion to EDS graphs.

A Few More Reflections on EDS

Sense Differentiation

- Focus on sentence meaning: dinstinguish grammaticalized contrasts:
- *look up the answer* (`_look_v_up`) vs. *look up the hill* (`_look_v_1`);
- *he broke the vase* (`_break_v_cause`) vs. *the vase broke* (`_break_v_1`);
- no grammatical contrast in *draw a house* vs. *draw a cart* (`_draw_v_1`).

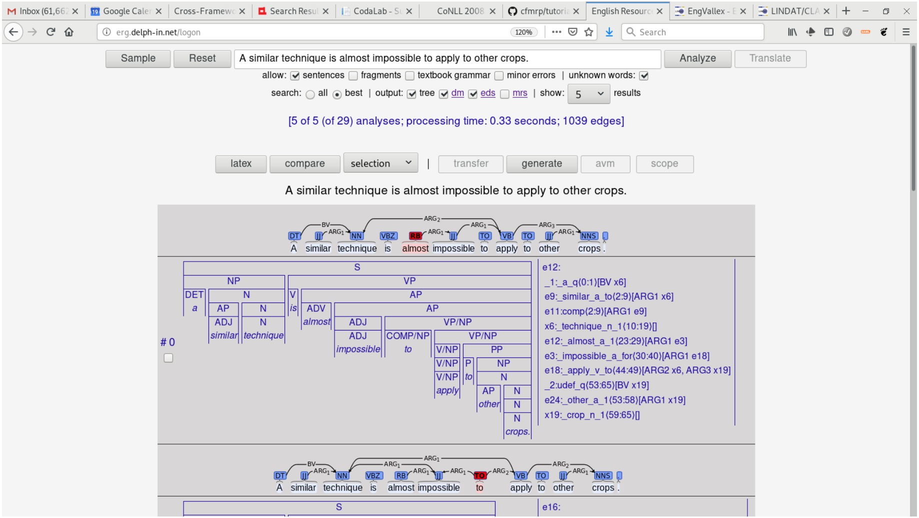
Limitations

- Grammar-based annotation: no 'correct' analysis in 5–15 % of inputs;
- limited resources for other languages (German, Japanese, Spanish, . . .);
- tense, aspect, mood, number, etc. as 'morphological' node properties;
- partial information about scope discarded in conversion to EDS graphs.

Dependency Minimal Recursion Semantics (DMRS)

- Recall: Original ERSs contain partial, underspecified scope information;
- Copestake (2009) monotonically extends EDS with scopal 'overlays'.

Background: English Resource Semantics On-Line

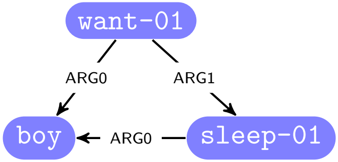


http://erg.delph-in.net/

Abstract Meaning Representation (AMR)

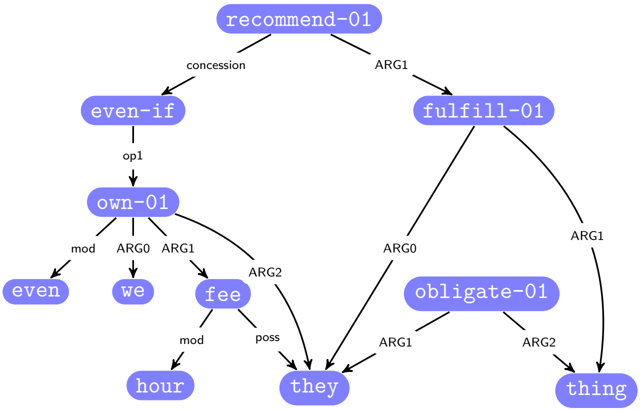
- Goals:
  - Capture predicate-argument structure of a sentence.
  - Nodes annotated with lexically decomposed predicates, using PropBank senses.
  - Different sentences with same meaning should have the same AMR.
  - Use for NLU, NLG, machine translation.
- First large-scale hand-annotated sembank:
  - "Little Prince" pilot annotation, ~1500 sentences
  - AMRBank v1 (LDC2014T12), ~13k sentences
  - AMRBank v2 (LDC2017T10), ~40k sentences, includes v1
  - ISI, since 2013 (Banarescu et al., 2013)
- Inter-annotator agreement: Smatch score around 70 on web text.

Predicate–Argument Structure in AMR



The boy wants to sleep.

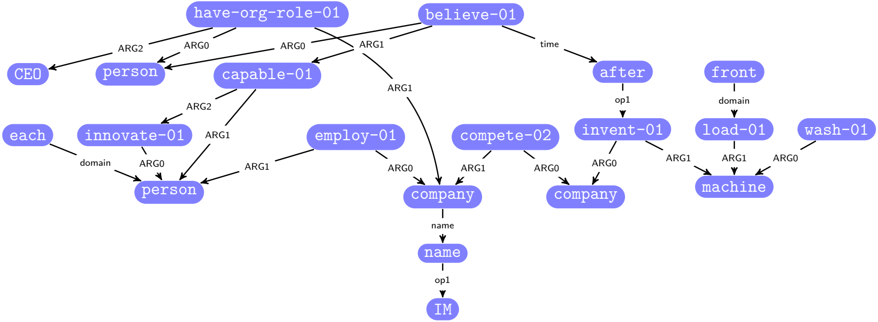
More Complex AMRs



Even if we owed their hourly fees, they still should fulfill their obligations.
(AMR2015 #34, simplified)

(a) implicit arguments
(b) raising-style argument passing
(c) lexical decomposition
(d) coreference

(Even) More Complex AMRs

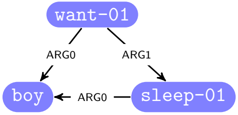


After its competitor invented the front loading washing machine, the CEO of the IM company believed that each of its employees had the ability for innovation. (AMR2015 #1, simplified)

(a) lexical decomposition
(b) named entities
(c) coreference
(d) quantification (?)

'Inverse Edges' in AMRs

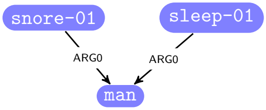
- Standard string format for AMRs is Penman-style:



The boy wants to sleep.

```
(w / want-01
   :ARG0 (b / boy)
   :ARG1 (s / sleep-01 :ARG0 b))
```

- String representation is based on DFS traversal of AMR, which sometimes traverses edges backwards. Represent with "label-of" edges:



The man who sleeps snores.

```
(sn / snore-01
   :ARG0 (m / man
     :ARG0-of (sl / sleep-01)))
```

- "label-of" edges are primarily artifacts of the string encoding. Some people interpret them as linguistically meaningful.

Limitations of AMR

Coreference

- Coreference-based edges indistinguishable from others.
- Linguistically, coreference is very different than e.g. control, namely noncompositional.
- Challenge for composition-based semantic parsers.

Expressive Capacity

- Operators such as quantifiers and modal verbs have scope. This is hard to represent when the MR is not a tree.
- People are stil trying.
- AMR has no model theory. "Man" and "every" are the same type of node label. If "man" refers to a set of men in the world, what's an "every"?

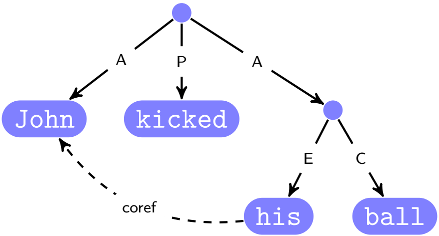
Universal Conceptual Cognitive Annotation (UCCA)

- Goals:
  - Capture predicate-argument structure of a sentence, in a way that abstracts over syntactic details.
  - Inspired by typological principles (Basic Linguistic Theory).
  - Make annotation as intuitive as possible, also cross-linguistically.
- Basic ideas:
  - Backbone of UCCA graph is a tree with the tokens as leaves and additional internal nodes, connected by a small set of semantic relations.
  - Additional remote edges represent argument sharing.
  - Multiple annotation layers, e.g. pred-arg structure vs. coreference.
- Annotations of "20000 Miles under the Sea" available in English, French, German; also web texts annotated.
- Hebrew University, since 2013 (Abend & Rappoport, 2013)
- Inter-annotator agreement: around 80 f-score.

UCCA Terminology

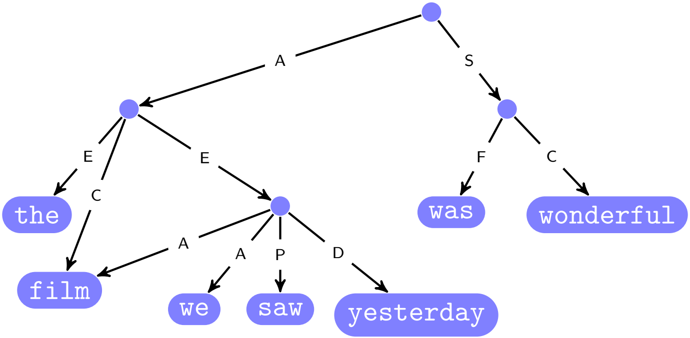
- A scene is a description of a single action or state. Sentences can contain multiple scenes. UCCA annotations distinguish between "processes" and "states".
- Scenes can have participants (≈ arguments) and adverbials/times (≈ modifiers).
- Below the clause level, distinguish centers from their elaborators and combine them with connectors.

UCCA: Basic Example



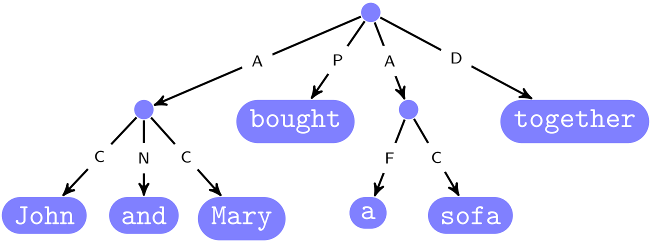
John kicked his ball.

UCCA: Argument sharing



The film we saw yesterday was wonderful.

UCCA: Coordination



John and Mary bought a sofa together.

Limitations of UCCA

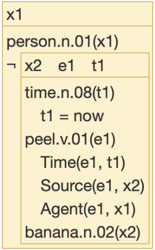
- Less well-studied approach than the others;
  may change with MRP Shared Task at CoNLL 2019.
- Does not distinguish different argument roles.
- Modification of a head causes major changes to graph structure;
  may be challenging for accurate parsing.

Graphbank Statistics

			CCD	DM	PSD	EDS	AMR	AMR^{-1}
COUNTS	(01)	number of graphs	39604	35656	35656	35656	10309	10309
	(02)	average number of tokens	23.47	22.51	22.51	22.51	20.62	20.62
	(03)	average number of nodes per token	0.88	0.77	0.64	0.99	0.67	0.67
	(04)	number of edge labels	6	59	90	10	135	100
TREENESS	(05)	$\%_g$ trees	1.45	2.31	42.26	0.98	52.48	18.60
	(06)	$\%_g$ treewidth one	29.27	69.82	43.08	65.37	52.72	52.72
	(07)	average treewidth	1.742	1.303	1.614	1.352	1.524	1.524
	(08)	maximal treewidth	5	3	7	3	4	4
	(09)	average edge density	1.070	1.019	1.073	1.047	1.065	1.065
	(10)	$\%_n$ reentrant	28.09	27.43	11.41	28.42	5.23	18.95
	(11)	$\%_g$ cyclic	1.28	0.00	0.00	0.04	3.15	0.71
	(12)	$\%_g$ not connected	12.53	6.57	0.70	1.49	0.00	0.00
	(13)	$\%_g$ multi-rooted	99.67	99.49	99.33	98.75	0.00	77.50
	(14)	percentage of non-top roots	47.78	44.94	4.34	41.15	0.00	19.39
ORDER	(15)	average edge length	2.582	2.684	3.320	–	–	–
	(16)	$\%_g$ noncrossing	48.23	69.21	64.61	–	–	–
	(17)	$\%_g$ pagenumber two	98.64	99.55	98.07	–	–	–
CONTROL	(01)	number of graphs	87	87	87	87	87	87
	(03)	average number of nodes per token	0.88	0.79	0.64	1.01	0.66	0.66
	(05)	$\%_g$ trees	1.15	1.15	45.98	1.15	60.92	3.45
	(06)	$\%_g$ treewidth one	37.93	81.61	47.13	81.61	60.92	60.92
	(07)	average treewidth	1.644	1.184	1.540	1.184	1.402	1.402
	(09)	average edge density	1.057	1.011	1.061	1.028	1.038	1.038
	(10)	$\%_n$ reentrant	28.92	27.73	10.28	27.77	2.88	21.09
	(11)	$\%_g$ cyclic	0.00	0.00	0.00	0.00	2.30	0.00
	(12)	$\%_g$ not connected	6.90	3.45	1.15	1.15	0.00	0.00
	(13)	$\%_g$ multi-rooted	100.00	100.00	100.00	98.85	0.00	93.10

(Kuhlmann & Oepen, 2016)

Parallel Meaning Bank



A person is not peeling a banana.

- Groningen Parallel Meaning Bank: 6k sentences in English, plus smaller corpora in DE, IT, NL (Basile et al., 2012; Abzianidze et al., 2017).
- First version: automatically annotated with Discourse Representation Structures (DRT) using Boxer.
- Iteratively hand-corrected, e.g. through game-based crowdsourcing.

Facets of Meaning in the Graphbanks

Type of Information	DM	PSD	EDS	DMRS	UCCA	AMR	PMB
Predicates–Arguments	$+$	$+$	$++$	$++$	$+$	$++$	$++$
Sense Differentiation	$+$	$++$	$+$	$+$	$-$	$++$	$++$
Scope & Quantification	$\pm$	$-$	$\pm$	$+$	$-$	$-$	$+$
Presupposition & Focus	$-$	$-$	$-$	$-$	$-$	$-$	$+$
Anaphoric Coreference	$-$	$-$	$-$	$-$	$-$	$+/-$	$+$
Grounding	$-$	$-$	$-$	$-$	$-$	$\pm$	$-$
...							

Facets of Meaning in the Graphbanks

Type of Information	DM	PSD	EDS	DMRS	UCCA	AMR	PMB
Predicates–Arguments	$+$	$+$	$++$	$++$	$+$	$++$	$++$
Sense Differentiation	$+$	$++$	$+$	$+$	$-$	$++$	$++$
Scope & Quantification	$\pm$	$-$	$\pm$	$+$	$-$	$-$[2]	$+$
Presupposition & Focus	$-$	$-$	$-$	$-$	$-$	$-$	$+$
Anaphoric Coreference	$-$	$-$[1]	$-$	$-$	$-$[3]	$+/-$	$+$
Grounding	$-$	$-$	$-$	$-$	$-$	$\pm$	$-$
...							

Represented in base resource, but not graphbank: [1]) Hajič et al. (2018).
Ongoing activities: [2]) Pustejovsky et al. (2019), [3]) Prange et al. (2019).

Facets of Natural Language Meaning

- Natural-language meaning has many facets.
- Each annotation scheme for semantics only captures some of these facets.
- Tradeoffs:
  - information captured by annotation
  - annotation effort
  - inter-annotator agreement

We focus here on graphbanks for English corpora.
We know of a handful of multilingual graphbanks:

framework	languages	corpora
AMR	ZH	Chinese AMRBank, 10k sentences `https://www.cs.brandeis.edu/~clp/camr/camr.` (Li et al., 2016)
AMR	IT, ES, DE, ZH	AMR projected across parallel corpora (Damonte & Cohen, 2018)
UCCA	FR, DE	20k Leagues under the Sea `http://www.cs.huji.ac.il/~oabend/ucca.html`
DRT	EN, DE, IT, NL	Groningen Parallel Meaning Bank `https://pmb.let.rug.nl/` (Abzianidze et al., 2017)

4 Semantic Parsing Approaches

Parsing to `Flavor (0)` Graphs

Flavor	Name	Example	Type of Anchoring
0	bilexical	SDP	nodes are sub-set of surface tokens
1	anchored	EDS	free node–sub-string correspondences
2	unanchored	AMR	no sub-string correspondences annotated

Parsing to `Flavor (0)` Graphs

Flavor	Name	Example	Type of Anchoring
0	bilexical	SDP	nodes are sub-set of surface tokens
1	anchored	EDS	free node–sub-string correspondences
2	unanchored	AMR	no sub-string correspondences annotated

Parsing to `Flavor (0)` graphs

- Nodes = tokens
- The goal is to predict labeled edges

Parsing to `Flavor (0)` Graphs

Flavor	Name	Example	Type of Anchoring
0	bilexical	SDP	nodes are sub-set of surface tokens
1	anchored	EDS	free node–sub-string correspondences
2	unanchored	AMR	no sub-string correspondences annotated

Parsing to `Flavor (0)` graphs

- Nodes = tokens
- The goal is to predict labeled edges

The drug was introduced in West Germany this year

Parsing to `Flavor (0)` Graphs

Flavor	Name	Example	Type of Anchoring
0	bilexical	SDP	nodes are sub-set of surface tokens
1	anchored	EDS	free node–sub-string correspondences
2	unanchored	AMR	no sub-string correspondences annotated

Parsing to `Flavor (0)` graphs

- Nodes = tokens
- The goal is to predict labeled edges

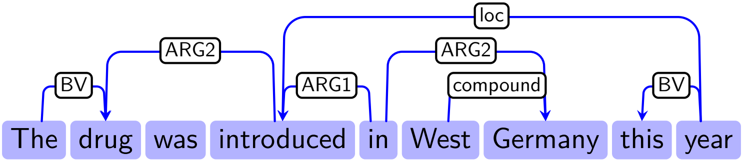
The drug was introduced in West Germany this year

Parsing to `Flavor (0)` Graphs

Flavor	Name	Example	Type of Anchoring
0	bilexical	SDP	nodes are sub-set of surface tokens
1	anchored	EDS	free node–sub-string correspondences
2	unanchored	AMR	no sub-string correspondences annotated

Parsing to `Flavor (0)` graphs

- Nodes = tokens
- The goal is to predict labeled edges



Semantic Parsing is Making Rapid Progress

	DM		PAS		PSD	
	id	ood	id	ood	id	ood
Du et al. (2015) (close)	89.1	81.8	91.3	87.2	75.7	73.3
H. Peng et al. (2017)	89.4	84.5	92.2	88.3	77.6	75.3
+Multitask learning	90.4	85.3	92.7	89.0	78.5	76.4
Dozat & Manning (2018)	93.7	88.9	94.0	90.8	81.0	79.4
Lindemann et al. (2019)	**93.9**	**90.3**	**94.5**	**92.5**	**82.0**	**81.5**
+Multitask learning	**94.1**	**90.5**	**94.7**	**92.8**	**82.1**	**81.6**

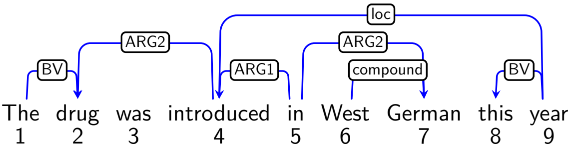
Accuracy of *edge prediction*

(id = in domain test set; ood = out of domain test set)

Semantic Parsing is Making Rapid Progress

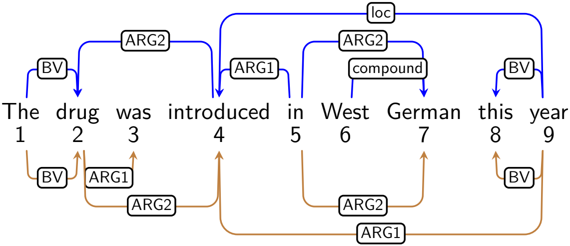
	DM		PAS		PSD	
	id	ood	id	ood	id	ood
Du et al. (2015) (close)	89.1	81.8	91.3	87.2	75.7	73.3
H. Peng et al. (2017)	89.4	84.5	92.2	88.3	77.6	75.3
+Multitask learning	90.4	85.3	92.7	89.0	78.5	76.4
Dozat & Manning (2018)	93.7	88.9	94.0	90.8	81.0	79.4
Lindemann et al. (2019)	**93.9**	**90.3**	**94.5**	**92.5**	**82.0**	**81.5**
+Multitask learning	**94.1**	**90.5**	**94.7**	**92.8**	**82.1**	**81.6**

Quantifying graph similarity is challenging.
What do these numbers mean?



$E_{\text{gold}} = \{(1, 2, \text{BV}), (2, 4, \text{ARG1}), \cdots\}$ $|E_{\text{gold}}| = 7$

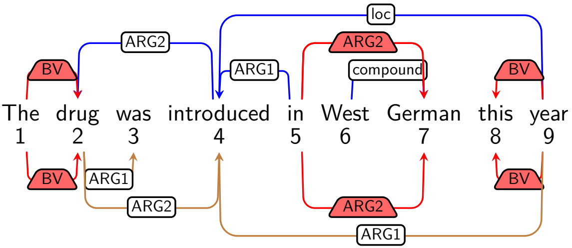
Evaluation for Parsing to `Flavor (0)` Graphs



$E_{\text{gold}} = \{(1, 2, \text{BV}), (2, 4, \text{ARG1}), \cdots\}$ $\quad |E_{\text{gold}}| = 7$

$E_{\text{system}} = \{(1, 2, \text{BV}), (2, 3, \text{ARG1}), \cdots\}$ $\quad |E_{\text{system}}| = 6$

Evaluation for Parsing to `Flavor (0)` Graphs

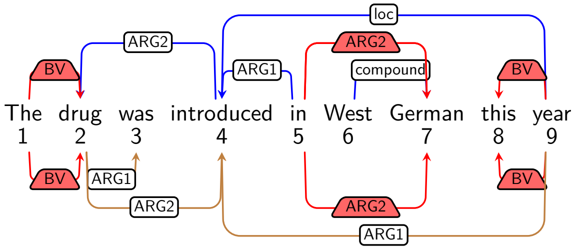


$E_{\text{gold}} = \{(1, 2, \text{BV}), (2, 4, \text{ARG1}), \cdots\}$ $|E_{\text{gold}}| = 7$

$E_{\text{system}} = \{(1, 2, \text{BV}), (2, 3, \text{ARG1}), \cdots\}$ $|E_{\text{system}}| = 6$

$E_{\text{match}} = E_{\text{gold}} \cap E_{\text{system}} = \{(1, 2, \text{BV}), (5, 7, \text{ARG1}), \cdots\}$ $|E_{\text{match}}| = 3$

Evaluation for Parsing to Flavor (0) Graphs



$E_{\text{gold}} = \{(1, 2, \text{BV}), (2, 4, \text{ARG1}), \cdots\}$ $\quad |E_{\text{gold}}| = 7$

$E_{\text{system}} = \{(1, 2, \text{BV}), (2, 3, \text{ARG1}), \cdots\}$ $\quad |E_{\text{system}}| = 6$

$E_{\text{match}} = E_{\text{gold}} \cap E_{\text{system}} = \{(1, 2, \text{BV}), (5, 7, \text{ARG1}), \cdots\}$ $\quad |E_{\text{match}}| = 3$

Precision	Recall	F-score
$\frac{\lvert E_{\text{match}}\rvert}{\lvert E_{\text{system}}\rvert} = 0.43$	$\frac{\lvert E_{\text{match}}\rvert}{\lvert E_{\text{gold}}\rvert} = 0.5$	$\frac{2 * \lvert E_{\text{match}}\rvert}{\lvert E_{\text{gold}}\rvert + \lvert E_{\text{system}}\rvert} = 0.46$

Magic Numbers Again

	DM		PAS		PSD	
	id	ood	id	ood	id	ood
Du et al. (2015) (close)	89.1	81.8	91.3	87.2	75.7	73.3
H. Peng et al. (2017)	89.4	84.5	92.2	88.3	77.6	75.3
+Multitask learning	90.4	85.3	92.7	89.0	78.5	76.4
Dozat & Manning (2018)	93.7	88.9	94.0	90.8	81.0	79.4
Lindemann et al. (2019)	**93.9**	**90.3**	**94.5**	**92.5**	**82.0**	**81.5**
+Multitask learning	**94.1**	**90.5**	**94.7**	**92.8**	**82.1**	**81.6**

Modern graph parsers are cool!

Parsing to `Flavor (1)` and `Flavor (2)` Graphs

Flavor	Name	Example	Type of Anchoring
0	bilexical	SDP	nodes are sub-set of surface tokens
1	anchored	EDS	free node–sub-string correspondences
2	unanchored	AMR	no sub-string correspondences annotated

Parsing to `Flavor (1)` and `Flavor (2)` graphs

- We need to predict labeled nodes

Parsing to `Flavor (1)` and `Flavor (2)` Graphs

Flavor	Name	Example	Type of Anchoring
0	bilexical	SDP	nodes are sub-set of surface tokens
1	anchored	EDS	free node–sub-string correspondences
2	unanchored	AMR	no sub-string correspondences annotated

Parsing to `Flavor (1)` and `Flavor (2)` graphs

- We need to predict labeled nodes and

`_the_q` `_introduce_v_to` `_year_n_1`

`_this_q_dem`

`_drug_n_1` `_in_p` `loc_nonsp` `named("Germany")`

`proper_q`

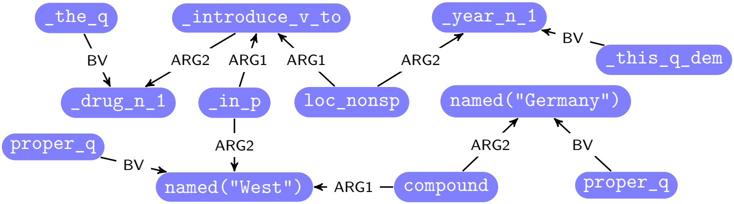
`named("West")` `compound` `proper_q`

Parsing to Flavor (1) and Flavor (2) Graphs

Flavor	Name	Example	Type of Anchoring
0	bilexical	SDP	nodes are sub-set of surface tokens
1	anchored	EDS	free node–sub-string correspondences
2	unanchored	AMR	no sub-string correspondences annotated

Parsing to Flavor (1) and Flavor (2) graphs

- We need to predict labeled nodes and labeled edges



Semantic Parsing is Making Rapid Progress

	EDS		**AMR 2015**	**AMR 201**
	Smatch F	EDM_{na}	Smatch F	Smatch F
Groschwitz et al. (2018)	-	-	70.2	71.0
Lyu & Titov (2018)	-	-	73.7	74.4
S. Zhang et al. (2019)	-	-	-	**76.3**
Buys & Blunsom (2017)	85.5	85.9	60.1	-
Chen, Sun, & Wan (2018)	**90.9**	**90.4**	-	-
Lindemann et al. (2019)	90.1	84.9	**74.3**	75.3
+ Multitask learning	90.4	85.2	74.5	75.3

Accuracy of *node and edge prediction*

Semantic Parsing is Making Rapid Progress

	EDS		AMR 2015	AMR 201
	Smatch F	EDM_{na}	Smatch F	Smatch F
Groschwitz et al. (2018)	-	-	70.2	71.0
Lyu & Titov (2018)	-	-	73.7	74.4
S. Zhang et al. (2019)	-	-	-	**76.3**
Buys & Blunsom (2017)	85.5	85.9	60.1	-
Chen, Sun, & Wan (2018)	**90.9**	**90.4**	-	-
Lindemann et al. (2019)	90.1	84.9	**74.3**	75.3
+ Multitask learning	90.4	85.2	74.5	75.3

Quantifying graph similarity is challenging.
What do these numbers mean?

Evaluation for Parsing to `Flavor (1)` Graphs

$_0$ *The* $_1$ *boy* $_2$ *wants* $_3$ *to* $_4$ *go* $_5$. $_6$

Evaluation for Parsing to `Flavor (1)` Graphs

$_0$ *The* $_1$ *boy* $_2$ *wants* $_3$ *to* $_4$ *go* $_5$. $_6$

`_the_q`$\langle 0, 1\rangle$ `_want_v_to`$\langle 2, 3\rangle$

`_boy_n_1`$\langle 1, 2\rangle$ `_go_v_1`$\langle 4, 5\rangle$

Evaluation for Parsing to `Flavor (1)` Graphs

$_0$ *The* $_1$ *boy* $_2$ *wants* $_3$ *to* $_4$ *go* $_5$. $_6$

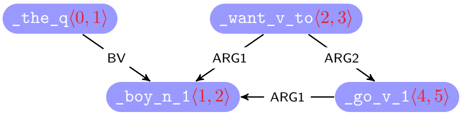
`_the_q`$\langle 0,1\rangle$ `_want_v_to`$\langle 2,3\rangle$

`_boy_n_1`$\langle 1,2\rangle$ `_go_v_1`$\langle 4,5\rangle$

$V_{\text{gold}} = \{(\langle 0,1\rangle, \texttt{\_the\_q}), \cdots\}$, $|V_{\text{gold}}| = 4$

Evaluation for Parsing to `Flavor (1)` Graphs

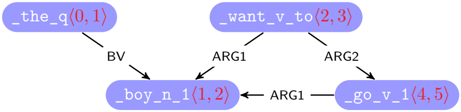
$_0$ *The* $_1$ *boy* $_2$ *wants* $_3$ *to* $_4$ *go* $_5$. $_6$



$V_{\text{gold}} = \{(\langle 0, 1 \rangle, \texttt{\_the\_q}), \cdots\}$, $|V_{\text{gold}}| = 4$

Evaluation for Parsing to `Flavor (1)` Graphs

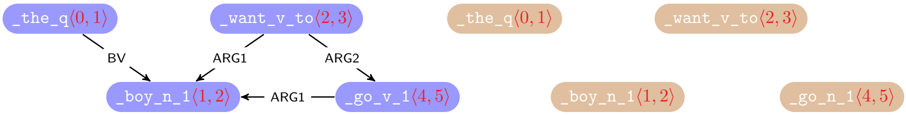
$_0$ *The* $_1$ *boy* $_2$ *wants* $_3$ *to* $_4$ *go* $_5$. $_6$



$V_{\text{gold}} = \{(\langle 0,1\rangle, \texttt{\_the\_q}), \cdots\},\ |V_{\text{gold}}| = 4$

$E_{\text{gold}} = \{(\langle 0,1\rangle, \text{BV}, \langle 1,2\rangle), \cdots\},\ |E_{\text{gold}}| = 4$

$_0$ *The* $_1$ *boy* $_2$ *wants* $_3$ *to* $_4$ *go* $_5$. $_6$

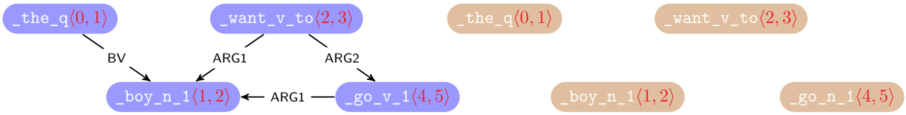


$V_{\text{gold}} = \{(\langle 0, 1\rangle, \texttt{\_the\_q}), \cdots\}, |V_{\text{gold}}| = 4$

$E_{\text{gold}} = \{(\langle 0, 1\rangle, \text{BV}, \langle 1, 2\rangle), \cdots\}, |E_{\text{gold}}| = 4$

Evaluation for Parsing to Flavor (1) Graphs

$_0$ *The* $_1$ *boy* $_2$ *wants* $_3$ *to* $_4$ *go* $_5$. $_6$

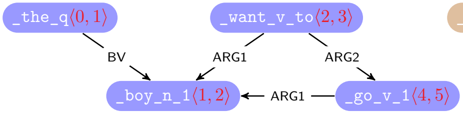


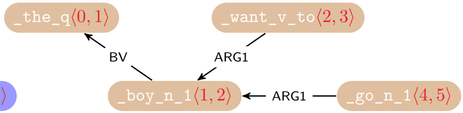
$V_{\text{gold}} = \{(\langle 0, 1\rangle, \texttt{\_the\_q}), \cdots\}$, $|V_{\text{gold}}| = 4$ $\quad V_{\text{sys}} = \{(\langle 0, 1\rangle, \texttt{\_the\_q}), \cdots\}$, $|V_{\text{sys}}| = 4$

$E_{\text{gold}} = \{(\langle 0, 1\rangle, \text{BV}, \langle 1, 2\rangle), \cdots\}$, $|E_{\text{gold}}| = 4$

Evaluation for Parsing to `Flavor (1)` Graphs

0 *The* 1 *boy* 2 *wants* 3 *to* 4 *go* 5 . 6



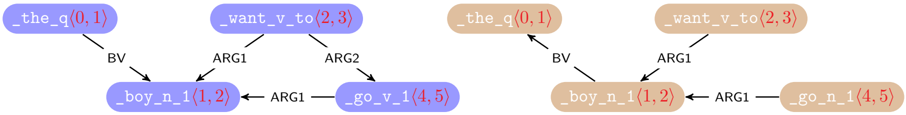


$V_{\text{gold}} = \{(\langle 0, 1\rangle, \texttt{\_the\_q}), \cdots\}$, $|V_{\text{gold}}| = 4$ $\qquad$ $V_{\text{sys}} = \{(\langle 0, 1\rangle, \texttt{\_the\_q}), \cdots\}$, $|V_{\text{sys}}| = 4$

$E_{\text{gold}} = \{(\langle 0, 1\rangle, \text{BV}, \langle 1, 2\rangle), \cdots\}$, $|E_{\text{gold}}| = 4$

Evaluation for Parsing to `Flavor (1)` Graphs

$_0$ *The* $_1$ *boy* $_2$ *wants* $_3$ *to* $_4$ *go* $_5$. $_6$

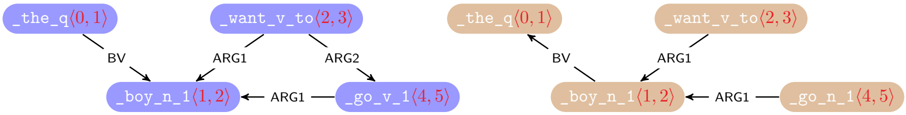


$V_{\text{gold}} = \{(\langle 0,1\rangle, \texttt{\_the\_q}), \cdots\}, |V_{\text{gold}}| = 4$ $\quad V_{\text{sys}} = \{(\langle 0,1\rangle, \texttt{\_the\_q}), \cdots\}, |V_{\text{sys}}| = 4$

$E_{\text{gold}} = \{(\langle 0,1\rangle, \text{BV}, \langle 1,2\rangle), \cdots\}, |E_{\text{gold}}| = 4$ $\quad E_{\text{sys}} = \{(\langle 1,2\rangle, \text{BV}, \langle 2,1\rangle), \cdots\}, |E_{\text{sys}}| = 3$

Evaluation for Parsing to Flavor (1) Graphs

0 *The* 1 *boy* 2 *wants* 3 *to* 4 *go* 5 . 6



$V_{\text{gold}} = \{(\langle 0, 1\rangle, \texttt{\_the\_q}), \cdots\}$, $|V_{\text{gold}}| = 4$ $\quad$ $V_{\text{sys}} = \{(\langle 0, 1\rangle, \texttt{\_the\_q}), \cdots\}$, $|V_{\text{sys}}| = 4$

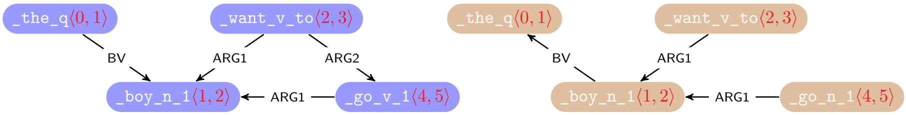
$E_{\text{gold}} = \{(\langle 0, 1\rangle, \text{BV}, \langle 1, 2\rangle), \cdots\}$, $|E_{\text{gold}}| = 4$ $\quad$ $E_{\text{sys}} = \{(\langle 1, 2\rangle, \text{BV}, \langle 2, 1\rangle), \cdots\}$, $|E_{\text{sys}}| = 3$

$$V_{\text{match}} = V_{\text{gold}} \cap V_{\text{sys}} = \{(\langle 1, 2\rangle, \texttt{\_boy\_n\_1}), \cdots\} \ |V_{\text{match}}| = 3$$

$$E_{\text{match}} = E_{\text{gold}} \cap E_{\text{sys}} = \{(\langle 2, 3\rangle, \text{ARG1}, \langle 1, 2\rangle), \cdots\} \ |E_{\text{match}}| = 2$$

Evaluation for Parsing to `Flavor (1)` Graphs

$_0$ *The* $_1$ *boy* $_2$ *wants* $_3$ *to* $_4$ *go* $_5$. $_6$



$V_{\text{gold}} = \{(\langle 0,1\rangle, \texttt{\_the\_q}), \cdots\},\ |V_{\text{gold}}| = 4$ $\quad$ $V_{\text{sys}} = \{(\langle 0,1\rangle, \texttt{\_the\_q}), \cdots\},\ |V_{\text{sys}}| = 4$

$E_{\text{gold}} = \{(\langle 0,1\rangle, \text{BV}, \langle 1,2\rangle), \cdots\},\ |E_{\text{gold}}| = 4$ $\quad$ $E_{\text{sys}} = \{(\langle 1,2\rangle, \text{BV}, \langle 2,1\rangle), \cdots\},\ |E_{\text{sys}}| = 3$

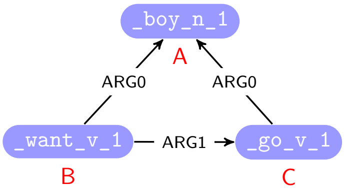
$$V_{\text{match}} = V_{\text{gold}} \cap V_{\text{sys}} = \{(\langle 1,2\rangle, \texttt{\_boy\_n\_1}), \cdots\}\ |V_{\text{match}}| = 3$$

$$E_{\text{match}} = E_{\text{gold}} \cap E_{\text{sys}} = \{(\langle 2,3\rangle, \text{ARG1}, \langle 1,2\rangle), \cdots\}\ |E_{\text{match}}| = 2$$

EDM_{n}	EDM_{a}	EDM_{na}
$\frac{2*\lvert V_{\text{match}}\rvert}{\lvert V_{\text{gold}}\rvert+\lvert V_{\text{sys}}\rvert} = 0.86$	$\frac{2*\lvert E_{\text{match}}\rvert}{\lvert E_{\text{gold}}\rvert+\lvert E_{\text{sys}}\rvert} = 0.57$	$\frac{2*(\lvert V_{\text{match}}\rvert+\lvert E_{\text{match}}\rvert)}{\lvert V_{\text{gold}}\rvert+\lvert V_{\text{sys}}\rvert+\lvert E_{\text{gold}}\rvert+\lvert E_{\text{sys}}\rvert} = 0.67$

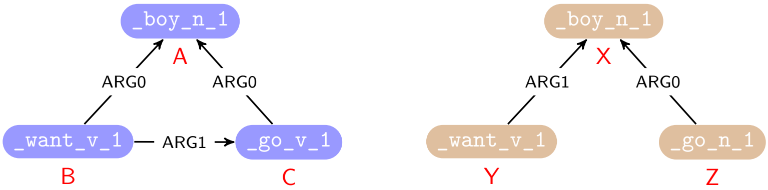
Evaluation for Parsing to `Flavor (2)` Graphs

Flavor	Name	Example	Type of Anchoring
1	anchored	EDS	free node–sub-string correspondences
2	unanchored	AMR	no sub-string correspondences annotated



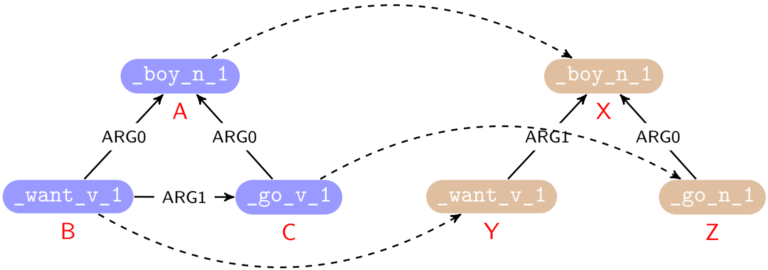
Evaluation for Parsing to `Flavor (2)` Graphs

Flavor	Name	Example	Type of Anchoring
1	anchored	EDS	free node–sub-string correspondences
2	unanchored	AMR	no sub-string correspondences annotated



Evaluation for Parsing to Flavor (2) Graphs

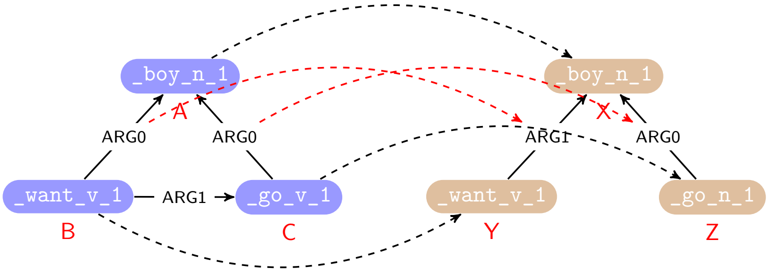
Flavor	Name	Example	Type of Anchoring
1	anchored	EDS	free node–sub-string correspondences
2	unanchored	AMR	no sub-string correspondences annotated



Assume an alignment: $A \leftrightarrow X$ & $B \leftrightarrow Y$ & $C \leftrightarrow Z$

Evaluation for Parsing to `Flavor (2)` Graphs

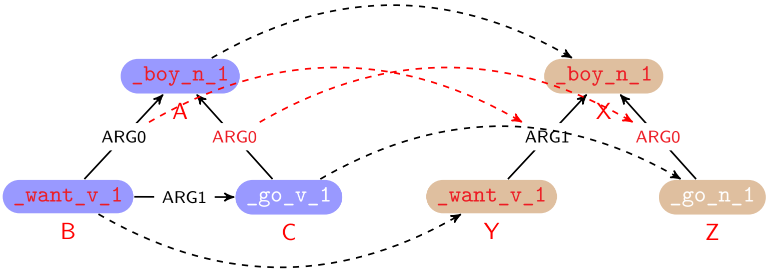
Flavor	Name	Example	Type of Anchoring
1	anchored	EDS	free node–sub-string correspondences
2	unanchored	AMR	no sub-string correspondences annotated



Assume an alignment: $A \leftrightarrow X$ & $B \leftrightarrow Y$ & $C \leftrightarrow Z$

Evaluation for Parsing to `Flavor (2)` Graphs

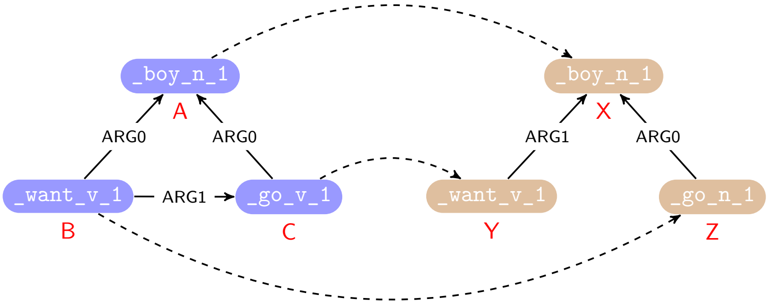
Flavor	Name	Example	Type of Anchoring
1	anchored	EDS	free node–sub-string correspondences
2	unanchored	AMR	no sub-string correspondences annotated



EDM_n	EDM_a	EDM_{na}
$\frac{2*2}{3+3} = 0.67$	$\frac{2*1}{3+2} = 0.40$	$\frac{2*3}{6+5} = 0.55$

Evaluation for Parsing to `Flavor (2)` Graphs

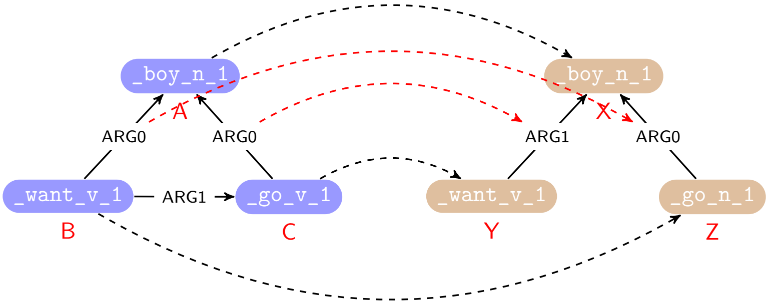
Flavor	Name	Example	Type of Anchoring
1	anchored	EDS	free node–sub-string correspondences
2	unanchored	AMR	no sub-string correspondences annotated



There are many such alignments: $A \leftrightarrow X$ & $B \leftrightarrow Z$ & $C \leftrightarrow Y$

Evaluation for Parsing to `Flavor (2)` Graphs

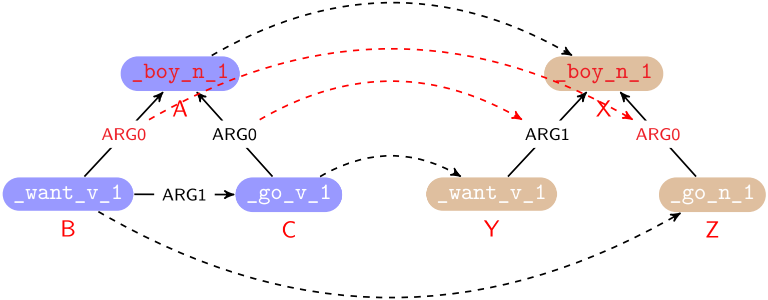
Flavor	Name	Example	Type of Anchoring
1	anchored	EDS	free node–sub-string correspondences
2	unanchored	AMR	no sub-string correspondences annotated



There are many such alignments: $A \leftrightarrow X$ & $B \leftrightarrow Z$ & $C \leftrightarrow Y$

Evaluation for Parsing to Flavor (2) Graphs

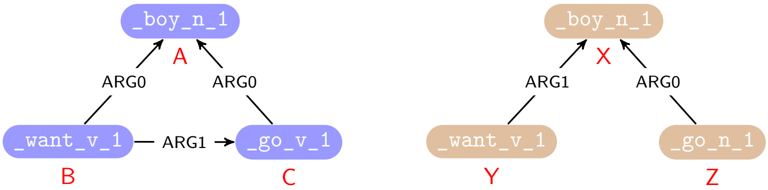
Flavor	Name	Example	Type of Anchoring
1	anchored	EDS	free node–sub-string correspondences
2	unanchored	AMR	no sub-string correspondences annotated



EDM_n	EDM_a	EDM_{na}
$\frac{2*1}{3+3} = 0.33$	$\frac{2*1}{3+2} = 0.40$	$\frac{2*2}{6+5} = 0.36$

Evaluation for Parsing to `Flavor (2)` Graphs

Flavor	Name	Example	Type of Anchoring
1	anchored	EDS	free node–sub-string correspondences
2	unanchored	AMR	no sub-string correspondences annotated



$$\textsc{Smatch}(G_g, G_s) = \max_{a \in \mathcal{A}(G_g, G_s)} \mathsf{EDM}_{\mathsf{na}}(a)$$

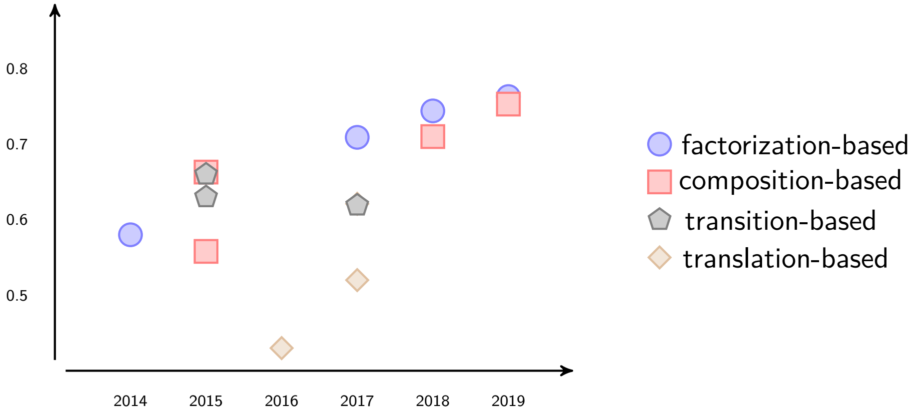
$\mathcal{A}(G_g, G_s)$ denotes the set of all plausible alignments between G_g and G_s

Magic Numbers Again

	EDS		AMR 2015	AMR 201
	Smatch F	EDM_{na}	Smatch F	Smatch F
Groschwitz et al. (2018)	-	-	70.2	71.0
Lyu & Titov (2018)	-	-	73.7	74.4
S. Zhang et al. (2019)	-	-	-	**76.3**
Buys & Blunsom (2017)	85.5	85.9	60.1	-
Chen, Sun, & Wan (2018)	**90.9**	**90.4**	-	-
Lindemann et al. (2019)	90.1	84.9	**74.3**	75.3
+Multitask learning	90.4	85.2	74.5	75.3

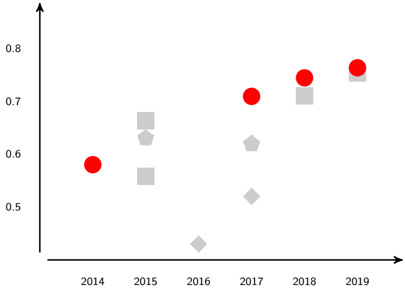
Modern graph parsers are cool!

Graph Parsing is Making Rapid Progress



AMR parsing accuracies reported in Flanigan et al. (2014); Foland & Martin (2017); Lyu & Titov (2018); S. Zhang et al. (2019); X. Peng et al. (2015); Artzi et al. (2015); Groschwitz et al. (2018); Lindemann et al. (2019); Barzdins & Gosko (2016); X. Peng et al. (2017); Konstas et al. (2017); Wang et al. (2015b, 2015a); Buys & Blunsom (2017)

Factorization-Based Approaches



- Inspired by the successful design of graph-based dependency parsers.
  Very early work: McDonald & Pereira (2006).
- Explicitly modeling the target structure.
- A parser knows how to evaluate the *goodness* of a candidate graph.
- A parser knows how to find the *best* graph from an extremely large set.

Sub-Tasks

the drug was introduced in West Germany this year

Sub-Tasks

the drug was introduced in West Germany this year

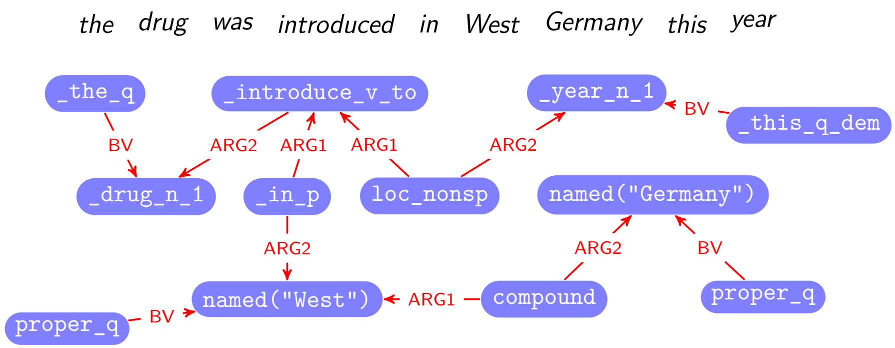
_the_q _introduce_v_to _year_n_1 _this_q_dem

_drug_n_1 _in_p loc_nonsp named("Germany")

named("West") compound proper_q

proper_q

Sub-Tasks



the drug was introduced in West Germany this year
_the_q
_introduce_v_to
_year_n_1
BV
_this_q_dem
BV
ARG2
ARG1
ARG1
ARG2
_drug_n_1
_in_p
loc_nonsp
named("Germany")
ARG2
ARG2
BV
named("West")
ARG1
compound
proper_q
proper_q
BV

Sub-Tasks

the drug was introduced in West Germany this year

`_the_q` `_introduce_v_to` `_year_n_1` `_this_q_dem`

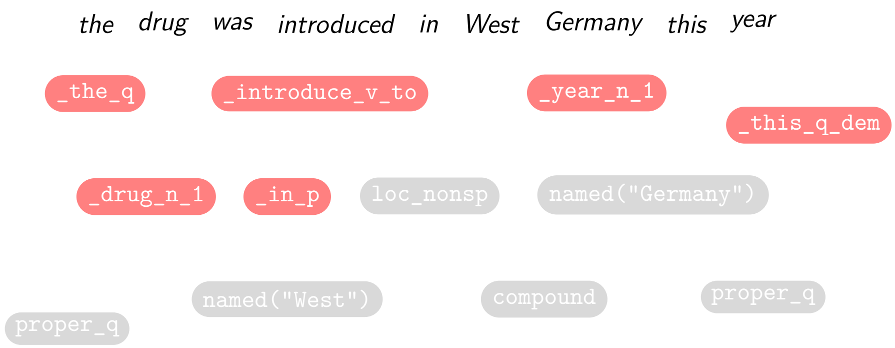
`_drug_n_1` `_in_p` `loc_nonsp` `named("Germany")`

`named("West")` `compound` `proper_q`

`proper_q`

Task 1: Concept Identification

Sub-Tasks



Task 1: Concept Identification

Sub-Tasks

the drug was introduced in West Germany this year

_the_q _introduce_v_to _year_n_1 _this_q_dem

_drug_n_1 _in_p loc_nonsp named("Germany")

named("West") compound proper_q

proper_q

Task 1: Concept Identification

Sub-Tasks

the drug was introduced in West Germany this year

_the_q _introduce_v_to _year_n_1 _this_q_dem

_drug_n_1 _in_p loc_nonsp named("Germany")

named("West") compound proper_q

proper_q

Task 1: Concept Identification

the drug was introduced in West Germany this year

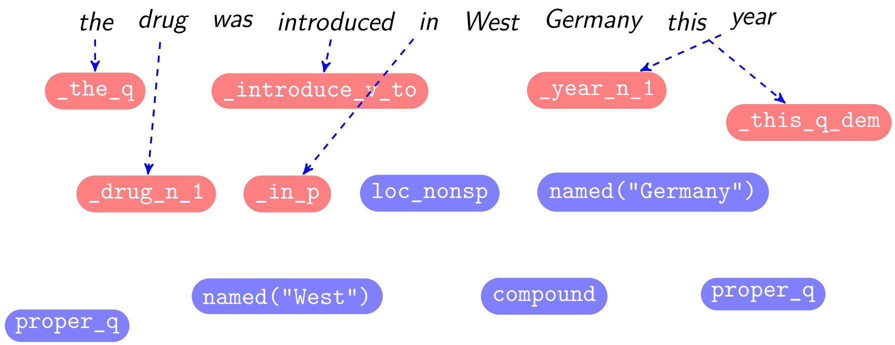
`_the_q` `_introduce_v_to` `_year_n_1` `_this_q_dem`

`_drug_n_1` `_in_p` `loc_nonsp` `named("Germany")`

`named("West")` `compound` `proper_q`

`proper_q`

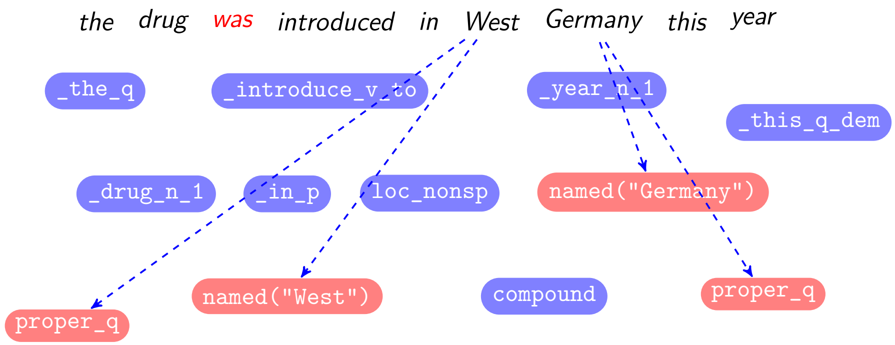
Task 1: Concept Identification



Task 0: Concept-to-word Alignment

Task 1: Concept Identification

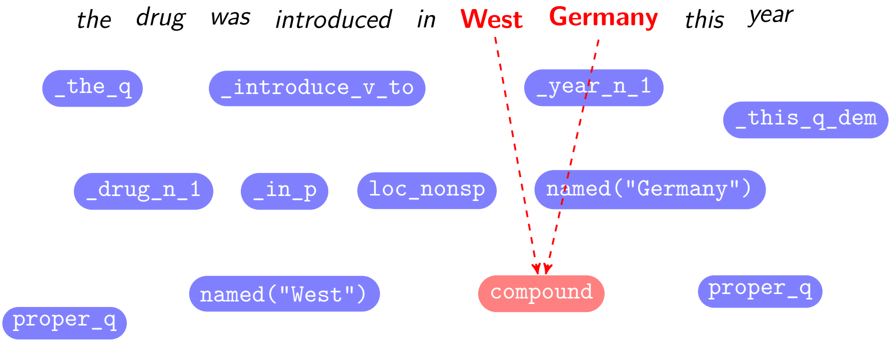
Sub-Tasks



Task 0: Concept-to-word Alignment

Task 1: Concept Identification

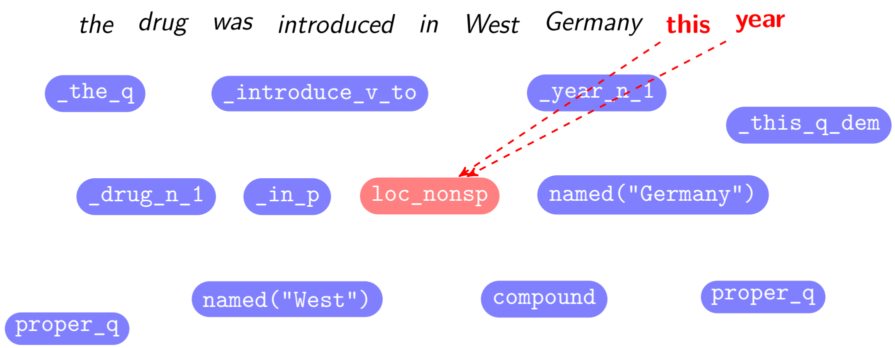
Sub-Tasks



Task 0: Concept-to-word Alignment

Task 1: Concept Identification

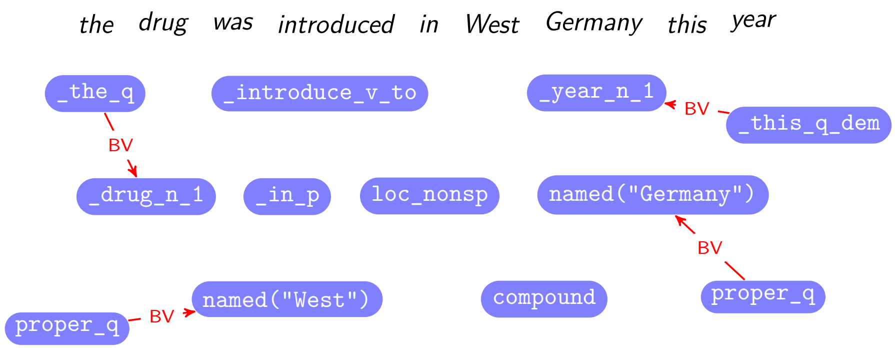
Sub-Tasks



Task 0: Concept-to-word Alignment

Task 1: Concept Identification

Sub-Tasks

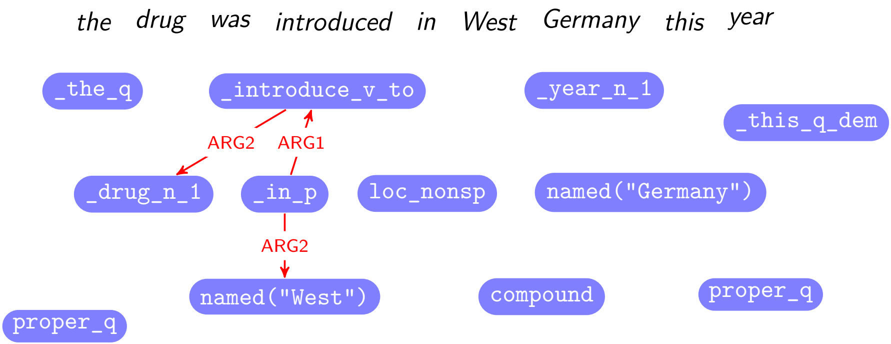


Task 0: Concept-to-word Alignment

Task 1: Concept Identification

Task 2: Relation Detection

Sub-Tasks

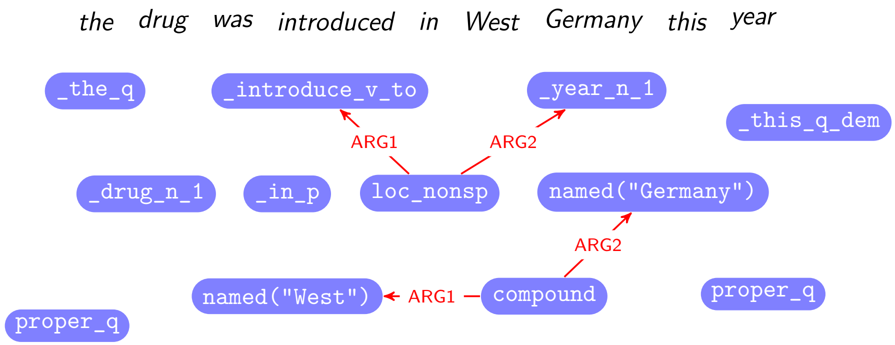


Task 0: Concept-to-word Alignment

Task 1: Concept Identification

Task 2: Relation Detection

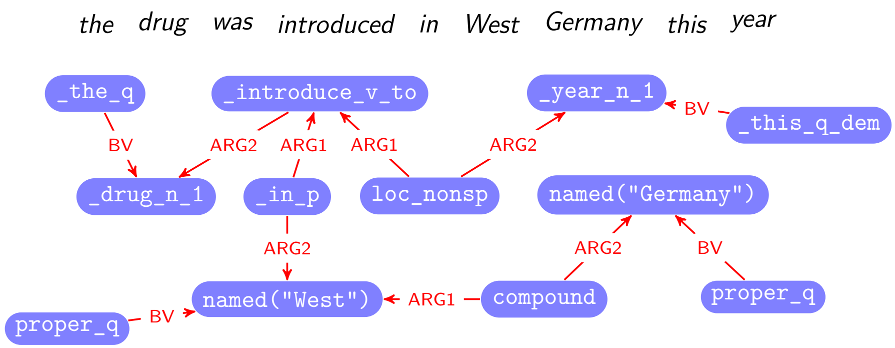
Sub-Tasks



Task 0: Concept-to-word Alignment

Task 1: Concept Identification

Task 2: Relation Detection



Task 0: Concept-to-word Alignment

Task 1: Concept Identification

Task 2: Relation Detection

Sub-Tasks

- Concept Identification (CI): predicting nodes
- Relation Detection (RD): linking nodes
- Concept-to-word alignment: finding concept-word correspondences

	Alignment	Concept Identification	Relation Detection
`Flavor (0)`			✔
`Flavor (1)`		✔	✔
`Flavor (2)`	✔	✔	✔

Sub-Tasks

- Concept Identification (CI): predicting nodes
- Relation Detection (RD): linking nodes
- Concept-to-word alignment: finding concept-word correspondences

	Alignment	Concept Identification	Relation Detection
`Flavor (0)`			✔
`Flavor (1)`		✔	✔
`Flavor (2)`	✔	✔	✔

As a Structured Prediction Problem

Maximum Subgraph Parsing

- **Start from** a directed graph $G = (V, E)$ that corresponds to $x = w_0, \ldots w_{n-1}$ and a score function that evaluates the *goodness* of a graph.
- **Search for** a subgraph $G' = (V, E' \subseteq E)$ that maximizes the score function:

$$G' = \arg \max_{G^*=(V,E^*\subseteq E)} \textsc{Score}(G^*)$$

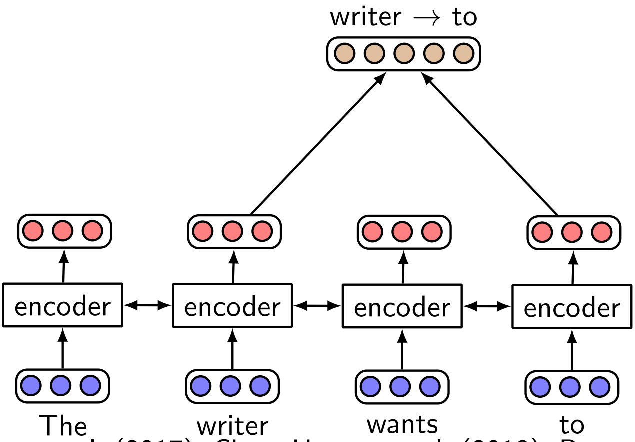
First-order factoriztion

$$G' = \arg \max_{G^*=(V,E^*\subseteq E)} \sum_{e \in E^*} \textsc{ScorePart}(e)$$

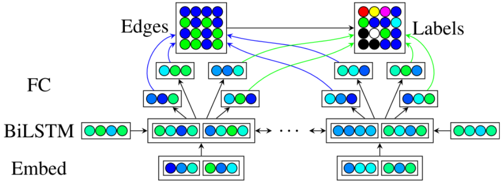
M. Kuhlmann and P. Jonsson. 2015. Parsing to Noncrossing Dependency Graphs.

Neural Scorers

$$G' = \arg \max_{G^* = (V, E^* \subseteq E)} \sum_{e \in E^*} \textsc{ScorePart}(e)$$



H. Peng et al. (2017); Chen, Huang, et al. (2018); Dozat & Manning (2018)



Concatenate word and POS tag embeddings.

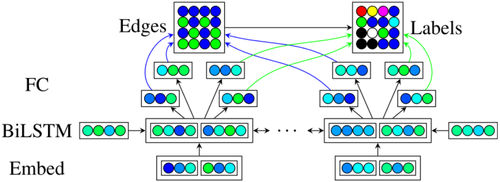
$$\mathbf{x}_i = \mathbf{e}_i^{\text{word}} \oplus \mathbf{e}_i^{\text{tag}}$$

Then *BiLSTM* them:

$$\mathbf{r}_1, \mathbf{r}_2, \ldots, \mathbf{r}_n = \textsc{BiLSTM}(\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n)$$

T. Dozat and C. Manning. 2018. Simpler but More Accurate Semantic Dependency Parsing.

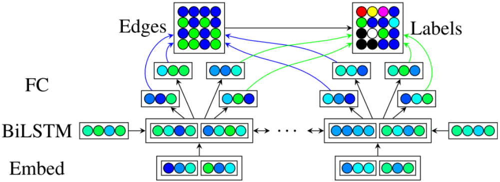
Neural Scorers



A word can be either a predicate or an argument. **Distinguish its grammartical/semantic function:**

$$
\begin{aligned}
\mathbf{h}_i^{\text{edge-head}} &= \text{FNN}^{\text{edge-head}}(\mathbf{r}_i) \\
\mathbf{h}_i^{\text{edge-dep}} &= \text{FNN}^{\text{edge-dep}}(\mathbf{r}_i) \\
\textsc{ScoreEdge}(s,i,j) &= \text{BIAFFINE}^{\text{edge}}(\mathbf{h}_i^{\text{edge-head}}, \mathbf{h}_j^{\text{edge-dep}})
\end{aligned}
$$

Neural Scorers

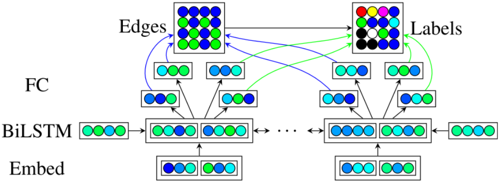


A word can be either a predicate or an argument. **Distinguish its grammartical/semantic function:**

$$\begin{aligned} \mathbf{h}_i^{\text{edge-head}} &= \text{FNN}^{\text{edge-head}}(\mathbf{r}_i) \\ \mathbf{h}_i^{\text{edge-dep}} &= \text{FNN}^{\text{edge-dep}}(\mathbf{r}_i) \\ \textsc{ScoreEdge}(s,i,j) &= \text{BIAFFINE}^{\text{edge}}(\mathbf{h}_i^{\text{edge-head}}, \mathbf{h}_j^{\text{edge-dep}}) \end{aligned}$$

FNN and BIAFFINE functions are popular

$$\begin{aligned} \text{BIAFFINE}(\mathbf{x}_1, \mathbf{x}_2) &= \mathbf{x}_1^\top \mathbf{U}\mathbf{x}_2 + \mathbf{W}(\mathbf{x}_1 \oplus \mathbf{x}_2) + \mathbf{b} \\ \text{FNN}(\mathbf{x}) &= \textsc{ReLU}(\mathbf{W}\mathbf{x} + \mathbf{b}) \end{aligned}$$



If $\textsc{ScoreEdge}(s, i, j) \geq 0$, then take $i \rightarrow j$ as an edge.

Select a label for $i \rightarrow j$ according to

$$\begin{aligned} \textsc{ScoreLabel}^{\text{label}}(i, j) &= \textsc{biaffine}^{\text{label}}(\mathbf{h}_i^{\text{label-head}}, \mathbf{h}_j^{\text{label-dep}}) \\ \mathbf{h}_i^{\text{label-head}} &= \textsc{fnn}^{\text{label-head}}(\mathbf{r}_i) \\ \mathbf{h}_i^{\text{label-dep}} &= \textsc{fnn}^{\text{label-dep}}(\mathbf{r}_i) \end{aligned}$$

Structure validation

Parsing to dependency trees

$$G' = \arg \max_{T^* = \mathcal{T}(G)} \sum_{e \in E^*} \textsc{ScorePart}(e)$$

Constraints on syntactic graphs: $\forall T' = (V, E') \in \mathcal{T}(G)$

- $E' \subseteq E$
- T' is a *directed tree*.

Constraints on semantic graphs

- Pagenumber-1 (= noncrossing): $O(n^3)$
- Pagenumber-2: NP-hard
- Pagenumber-2 and 1-endpoint-crossing and ...: $O(n^4)$

Kuhlmann & Jonsson (2015); Cao et al. (2017a, 2017b)

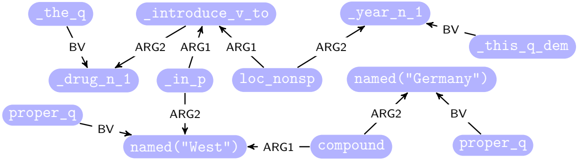
Sub-Tasks

- Concept Identification (CI): predicting nodes
- Relation Detection (RD): linking nodes
- Concept-to-word alignment: finding concept-word correspondences

	Alignment	Concept Identification	Relation Detection
`Flavor (0)`			✔
`Flavor (1)`		✔	✔
`Flavor (2)`	✔	✔	✔

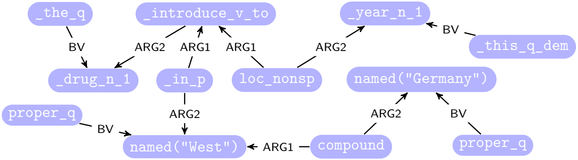
Concept Identification for `Flavor (1)` Graphs

Flavor	Name	Example	Type of Anchoring
1	anchored	EDS	free node–sub-string correspondences
2	unanchored	AMR	no sub-string correspondences annotated



Concept Identification for `Flavor (1)` Graphs

Flavor	Name	Example	Type of Anchoring
1	anchored	EDS	free node–sub-string correspondences
2	unanchored	AMR	no sub-string correspondences annotated



The	drug	was	introduced	in	West	Germany	this	year
_the_q	_drug_n_1	∅	_introduce_v_to	_in_p	named("W") proper_q	named("G") proper_q	_this_q_dem	_year_n_1
					compound		loc_nonsp	

Concept Identification for Flavor (1) Graphs

The	drug	was	introduced	in	West	Germany	this	year
_the_q	_drug_n_1	∅	_introduce_v_to	_in_p	named("W") proper_q	named("G") proper_q	_this_q_dem	_year_n_1
					compound		loc_nonsp	

Almost Sequence Labeling

- Some nodes are linked to sub-words.
- Some nodes are linked to multiple words.

Solutions

- Preprocessing: every node is assigned to a single word
- Chunking: joint segmentation and tagging
  - **B**-x: begin of x
  - **I**-x: inside x
- Lightweight phrase-structure parsing (UCCA)

Concept Identification for Flavor (1) Graphs

The	drug	was	introduced	in	West	Germany	this	year
_the_q	_drug_n_1	∅	_introduce_v_to	_in_p	named("W") proper_q	named("G") proper_q	_this_q_dem	_year_n_1
					compound			loc_nonsp

Almost Sequence Labeling

- Some nodes are linked to sub-words.
- Some nodes are linked to multiple words.

Solutions

- Preprocessing: every node is assigned to a single word
- Chunking: joint segmentation and tagging
  - **B**-x: begin of x
  - **I**-x: inside x
- Lightweight phrase-structure parsing (UCCA)

Concept Identification for `Flavor (1)` Graphs

The	drug	was	introduced	in	West	Germany	this	year
`_the_q`	`_drug_n_1`	$\emptyset$	`_introduce_v_to`	`_in_p`	`named("W")` `proper_q`	`named("G")` `proper_q`	`_this_q_dem`	`_year_n_1`
					`B-compound`	`I-compound`	`B-loc_nonsp`	`I-loc_nonsp`

Almost Sequence Labeling

- Some nodes are linked to sub-words.
- Some nodes are linked to multiple words.

Solutions

- Preprocessing: every node is assigned to a single word
- Chunking: joint segmentation and tagging
  - **B**-x: begin of x
  - **I**-x: inside x
- Lightweight phrase-structure parsing (UCCA)

Concept Identification for `Flavor (1)` Graphs

The	drug	was	introduced	in	West	Germany	this	year
`_the_q`	`_drug_n_1`	$\emptyset$	`_introduce_v_to`	`_in_p`	`named("W")` `proper_q`	`named("G")` `proper_q`	`_this_q_dem`	`_year_n_1`
					`compound`		`loc_nonsp`	

Almost Sequence Labeling

- Some nodes are linked to sub-words.
- Some nodes are linked to multiple words.

Solutions

- Preprocessing: every node is assigned to a single word
- Chunking: joint segmentation and tagging
  - **B**-x: begin of x
  - **I**-x: inside x
- Lightweight phrase-structure parsing (UCCA)

Neural Tagging

Almost Sequence Labeling

- Preprocessing: every node is assigned to a single word
- Chunking: joint segmentation and tagging

Challenge

Like POS tagging but with `thousands` of labels.

Delexicalization

The	drug	introduced	in	West	Germany	this	year

`_the_q`	`_drug_n_1`	`_introduce_v_to`	`_in_p`	`named("W")`
				`compound`
				`proper_q`
					`named("G")`
					`proper_q`
						`_this_q_dem`	`_year_n_1`
							`loc_nonsp`

Neural Tagging

Almost Sequence Labeling

- Preprocessing: every node is assigned to a single word
- Chunking: joint segmentation and tagging

Challenge

Like POS tagging but with thousands of labels.

Delexicalization

The	drug	introduced	in	West	Germany	this	year
_the_q	_drug_n_1	_introduce_v_to	_in_p	named("W") compound proper_q	named("G") proper_q	_this_q_dem	_year_n_1 loc_nonsp
*_q	*_n_1	*_v_to	*_p	named compound proper_q	named proper_q	*_q_dem	*_n_1 loc_nonsp

Concept Identification for `Flavor (1)` Graphs

Solution

Like POS tagging but with 1000+ labels.

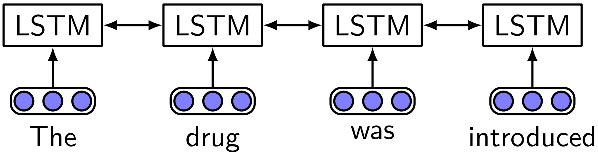
- Pretraining: ELMo, BERT, etc.
- Word encoder: LSTM, Transformer, etc.
- Classification

Concept Identification for `Flavor (1)` Graphs

Solution

Like POS tagging but with 1000+ labels.

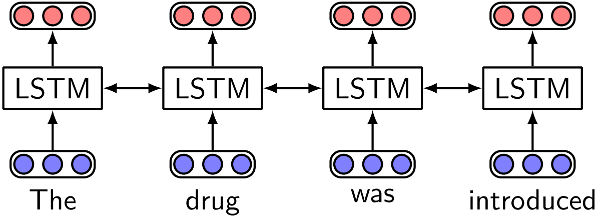
- Pretraining: ELMo, BERT, etc.
- Word encoder: LSTM, Transformer, etc.
- Classification



Solution

Like POS tagging but with 1000+ labels.

- Pretraining: ELMo, BERT, etc.
- Word encoder: LSTM, Transformer, etc.
- Classification

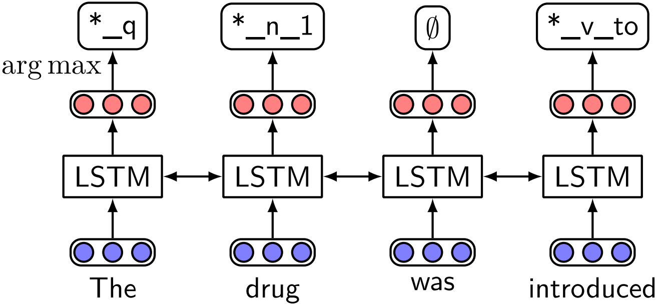


Concept Identification for `Flavor (1)` Graphs

Solution

Like POS tagging but with 1000+ labels.

- Pretraining: ELMo, BERT, etc.
- Word encoder: LSTM, Transformer, etc.
- Classification



Sub-Tasks

- Concept Identification (CI): predicting nodes
- Relation Detection (RD): linking nodes
- Concept-to-word alignment: finding concept-word correspondences

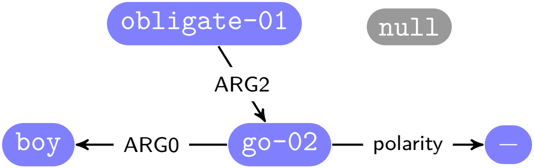
	Alignment	Concept Identification	Relation Detection
`Flavor (0)`			✔
`Flavor (1)`		✔	✔
`Flavor (2)`	✔	✔	✔

Transition- and composition-based approaches also need concept-to-word alignments.

Concept Identification for `Flavor (2)` graphs

the boys must not go

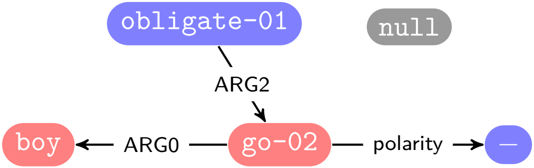
$\Downarrow$



Concept Identification for Flavor (2) graphs

the boys must not go

⇓



No annotations for concept-to-word alignment

- Heuristic rules (Flanigan et al., 2014)
- Linearize graphs and reuse word alignment tools, e.g. `GIZA++` and `BerkeleyAligner`, etc. (Pourdamghani et al., 2014)
- Consider all possible alignments (Lyu & Titov, 2018)

Concept Identification for `Flavor (2)` graphs

the boys must not go

$\Downarrow$

`obligate-01` `go-02` `boy` –

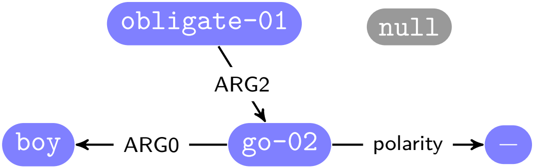
No annotations for concept-to-word alignment

- Heuristic rules (Flanigan et al., 2014)
- Linearize graphs and reuse word alignment tools, e.g. `GIZA++` and `BerkeleyAligner`, etc. (Pourdamghani et al., 2014)
- Consider all possible alignments (Lyu & Titov, 2018)

Concept Identification for `Flavor (2)` graphs

the boys must not go

$\Downarrow$

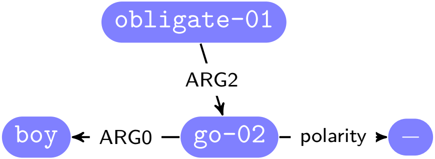


No annotations for concept-to-word alignment

- Heuristic rules (Flanigan et al., 2014)
- Linearize graphs and reuse word alignment tools, e.g. `GIZA++` and `BerkeleyAligner`, etc. (Pourdamghani et al., 2014)
- Consider all possible alignments (Lyu & Titov, 2018)

Concept Identification for Flavor (2) Graphs

the boys must not go

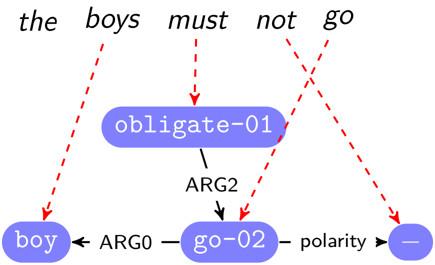


$$P_{\theta,\phi}(\mathbf{c}, R|\mathbf{w}) = \sum_{\mathbf{a}} P(\mathbf{a})P_{\theta}(\mathbf{c}|\mathbf{a}, \mathbf{w})P_{\phi}(R|\mathbf{a}, \mathbf{w}, \mathbf{c})$$

1. the concept identification model: $P_{\theta}(\mathbf{c}|\mathbf{a}, \mathbf{w})$
2. the relation identification model: $P_{\phi}(R|\mathbf{a}, \mathbf{w}, \mathbf{c})$
3. the alignment model: $Q_{\psi}(\mathbf{a}|\mathbf{c}, R, \mathbf{w})$

C. Lyu and I. Titov. 2018. AMR Parsing as Graph Prediction with Latent Alignment

Concept Identification for `Flavor (2)` Graphs

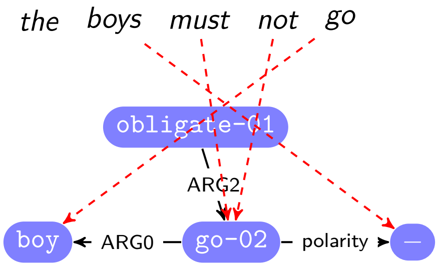


$$P_{\theta,\phi}(\mathbf{c}, R|\mathbf{w}) = \sum_{\mathbf{a}} P(\mathbf{a})P_\theta(\mathbf{c}|\mathbf{a},\mathbf{w})P_\phi(R|\mathbf{a},\mathbf{w},\mathbf{c})$$

1. the concept identification model: $P_\theta(\mathbf{c}|\mathbf{a},\mathbf{w})$
2. the relation identification model: $P_\phi(R|\mathbf{a},\mathbf{w},\mathbf{c})$
3. the alignment model: $Q_\psi(\mathbf{a}|\mathbf{c},R,\mathbf{w})$

C. Lyu and I. Titov. 2018. AMR Parsing as Graph Prediction with Latent Alignment

Concept Identification for `Flavor (2)` Graphs

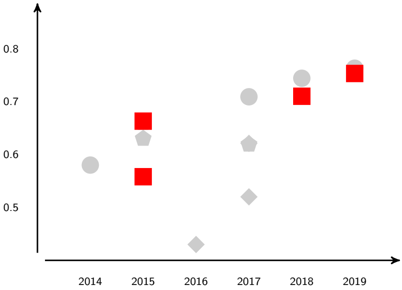


$$P_{\theta,\phi}(\mathbf{c}, R|\mathbf{w}) = \sum_{\mathbf{a}} P(\mathbf{a})P_\theta(\mathbf{c}|\mathbf{a}, \mathbf{w})P_\phi(R|\mathbf{a}, \mathbf{w}, \mathbf{c})$$

1. the concept identification model: $P_\theta(\mathbf{c}|\mathbf{a}, \mathbf{w})$
2. the relation identification model: $P_\phi(R|\mathbf{a}, \mathbf{w}, \mathbf{c})$
3. the alignment model: $Q_\psi(\mathbf{a}|\mathbf{c}, R, \mathbf{w})$

C. Lyu and I. Titov. 2018. AMR Parsing as Graph Prediction with Latent Alignment

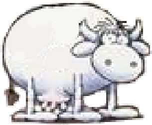
Composition-Based Approaches



- Inspired by *old school*, rule-based approaches.
- Explicitly modeling the syntactico-semantic derivation process.
- A parser knows how to evaluate the *goodness* of a derivation process.
- A parser knows how to find the *best* derivation from a large set of derivations that are licensed by a symbolic *system*.

Reminder: Compositionality

dotted $+$ **cow** $=$ **dotted cow**

(Example from Jan van Eijck)

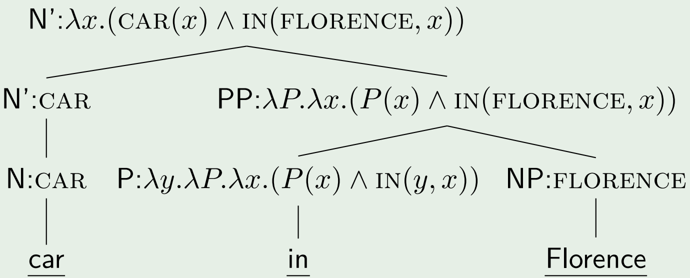
The Principle of Compositionality

> *The meaning of an expression is a function of the meanings of its parts and of the way they are syntactically combined.* B. Partee

Compositional Semantics

Tree construction + Lexical interpretation ⇒ Meaning representation

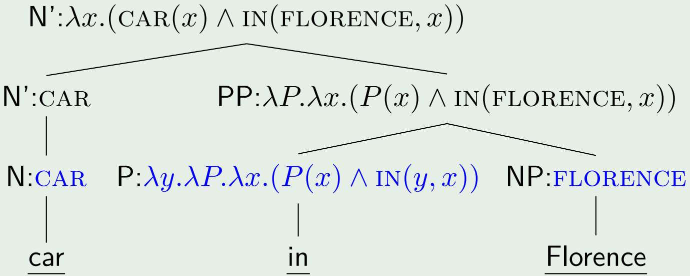
Using λ Expressions



Compositional Semantics

Tree construction + Lexical interpretation ⇒ Meaning representation

Using λ Expressions



Compositional Semantics

Tree construction + Lexical interpretation $\Rightarrow$ Meaning representation

Using λ Expressions

- N’:$\lambda x.(\textsc{car}(x) \wedge \textsc{in}(\textsc{florence}, x))$
  - N’:$\textsc{car}$
    - N:$\textsc{car}$
      - car
  - PP:$\lambda P.\lambda x.(P(x) \wedge \textsc{in}(\textsc{florence}, x))$
    - P:$\lambda y.\lambda P.\lambda x.(P(x) \wedge \textsc{in}(y, x))$
      - in
    - NP:$\textsc{florence}$
      - Florence

Compositional Semantics

Tree construction + Lexical interpretation ⇒ Meaning representation

Using λ Expressions

- N':$\lambda x.(\textsc{car}(x) \wedge \textsc{in}(\textsc{florence}, x))$
  - N':$\textsc{car}$
    - N:$\textsc{car}$
      - car
  - PP:$\lambda P.\lambda x.(P(x) \wedge \textsc{in}(\textsc{florence}, x))$
    - P:$\lambda y.\lambda P.\lambda x.(P(x) \wedge \textsc{in}(y, x))$
      - in
    - NP:$\textsc{florence}$
      - Florence

Compositional Semantics

Tree construction + Lexical interpretation ⇒ Meaning representation

Using λ Expressions

- N’:$\lambda x.(\textsc{car}(x) \land \textsc{in}(\textsc{florence}, x))$
  - N’:$\textsc{car}$
    - N:$\textsc{car}$
      - car
  - PP:$\lambda P.\lambda x.(P(x) \land \textsc{in}(\textsc{florence}, x))$
    - P:$\lambda y.\lambda P.\lambda x.(P(x) \land \textsc{in}(y, x))$
      - in
    - NP:$\textsc{florence}$
      - Florence

Manipulating Graphs

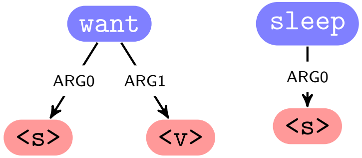
- Graph algebra (AM algebra)
- Graph grammar (hyperedge replacement grammar)

AM Algebra

S-Graph

An s-graph is a graph in which nodes (called sources) are marked with "public names".

Examples



"root" source

- want
- sleep

other sources

- <s> (for "subjects")
- <v> (for "verb complements")

AM Algebra

The writer wants to sleep soundly

AM algebra provides a systematic way to construct graphs

- term (=tree of operation symbols)
- value (=s-graph)

Two operations for combining s-graphs:

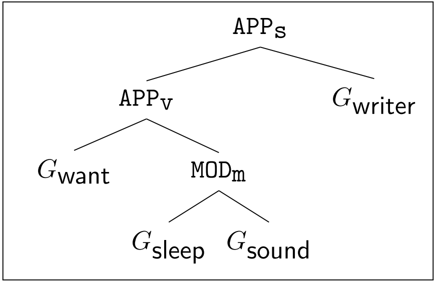
- Apply (=head+complement)
- Modify (=head+modifier)

Groschwitz et al. (2017)

AM Algebra

The writer wants to sleep soundly

$\Downarrow$



AM algebra provides a systematic way to construct graphs

- term (=tree of operation symbols)
- value (=s-graph)

Two operations for combining s-graphs:

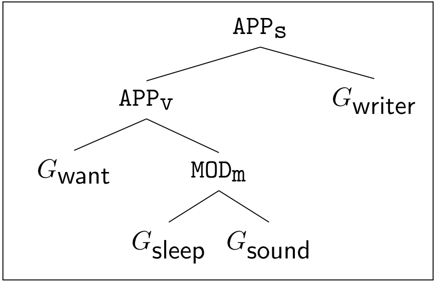
- Apply (=head+complement)
- Modify (=head+modifier)

Groschwitz et al. (2017)

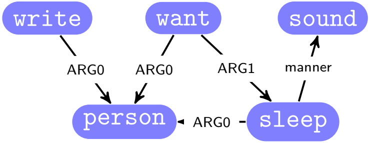
AM Algebra

The writer wants to sleep soundly

$\Downarrow$



$\Downarrow$



AM algebra provides a systematic way to construct graphs

- term (=tree of operation symbols)
- value (=s-graph)

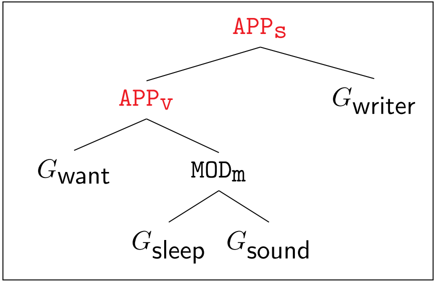
Two operations for combining s-graphs:

- Apply (=head+complement)
- Modify (=head+modifier)

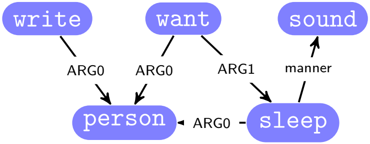
Groschwitz et al. (2017)

The writer wants to sleep soundly

⇓



⇓



AM algebra provides a systematic way to construct graphs

- term (=tree of operation symbols)
- value (=s-graph)

Two operations for combining s-graphs:

- Apply (=head+complement)
- Modify (=head+modifier)

Groschwitz et al. (2017)

AM algebra provides a systematic way to construct graphs

- term (=tree of operation symbols)
- value (=s-graph)

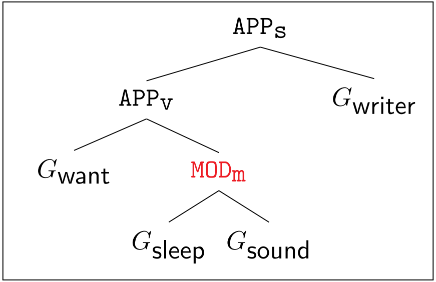
Two operations for combining s-graphs:

- Apply (=head+complement)
- Modify (=head+modifier)

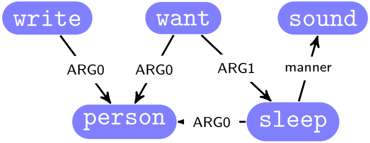
Groschwitz et al. (2017)

The writer wants to sleep soundly

$\Downarrow$



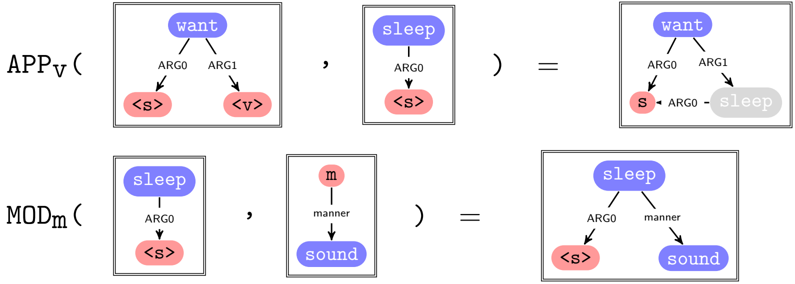
$\Downarrow$



AM Algebra

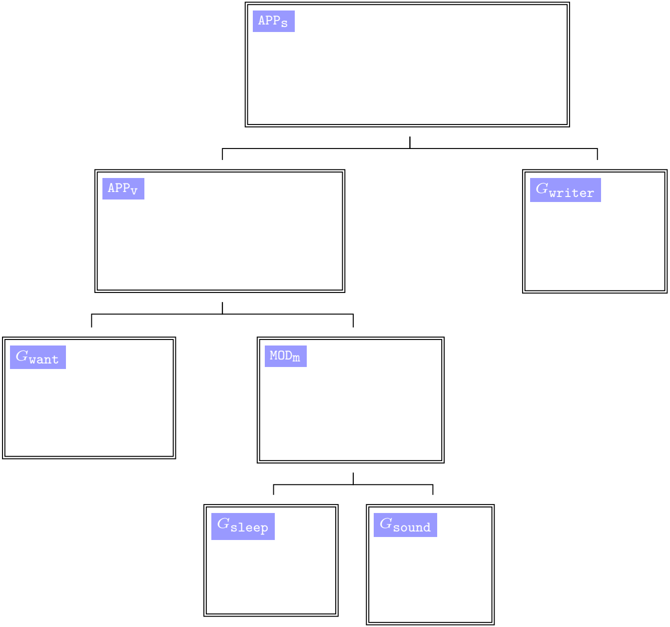
Two operations for combining s-graphs:

Apply (=head+complement), Modify (=head+modifier)



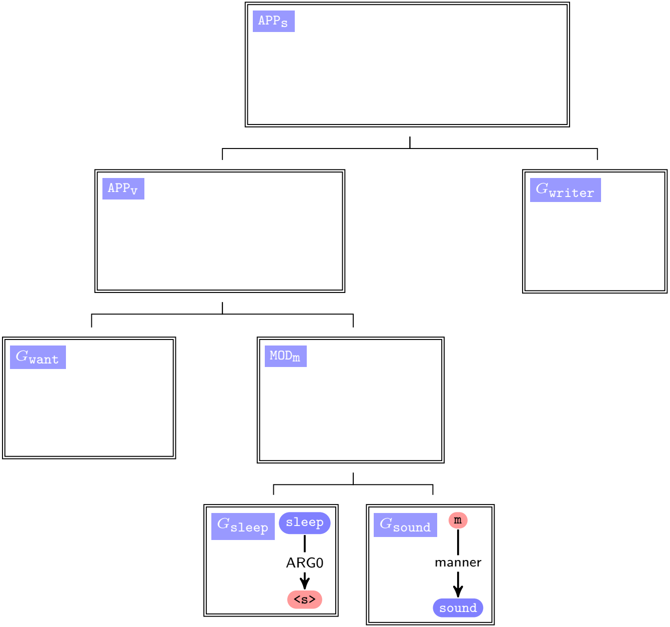
J. Groschwitz, M. Fowlie, M. Johnson and A. Koller. 2017. A constrained graph algebra for semantic parsing with AMRs.

AM Algebra



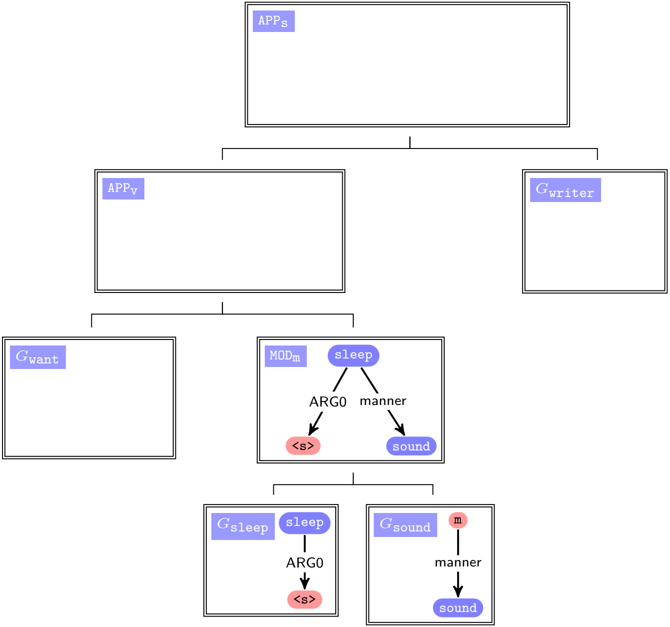
$\mathtt{APP_s}$
$\mathtt{APP_v}$
$G_{\mathtt{writer}}$
$G_{\mathtt{want}}$
$\mathtt{MOD_m}$
$G_{\mathtt{sleep}}$
$G_{\mathtt{sound}}$

AM Algebra



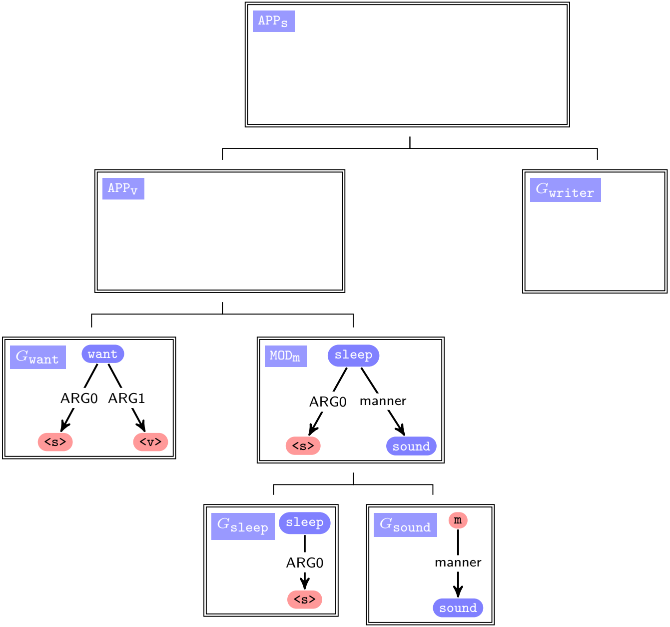
$\mathrm{APP_s}$
$\mathrm{APP_v}$
$G_{\mathtt{writer}}$
$G_{\mathtt{want}}$
$\mathrm{MOD_m}$
$G_{\mathtt{sleep}}$
sleep
ARG0
<s>
$G_{\mathtt{sound}}$
m
manner
sound

AM Algebra



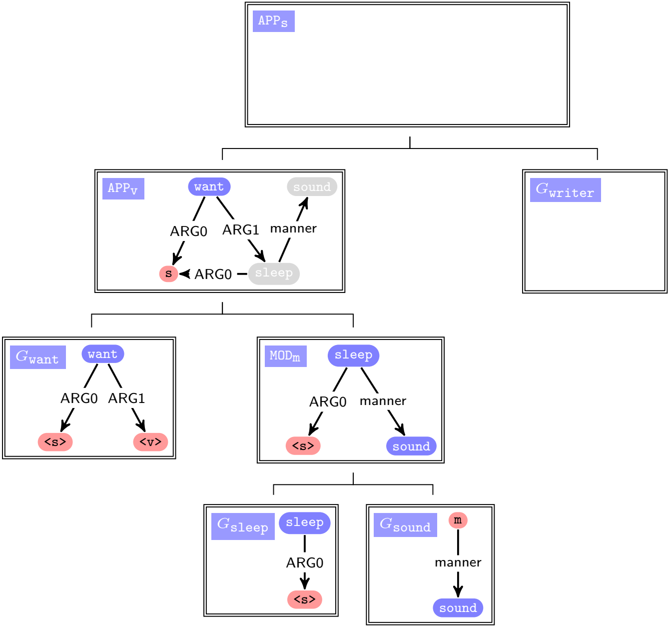
APPs
APPv
Gwriter
Gwant
MODm
sleep
ARG0
manner
<s>
sound
Gsleep
sleep
ARG0
<s>
Gsound
m
manner
sound

AM Algebra



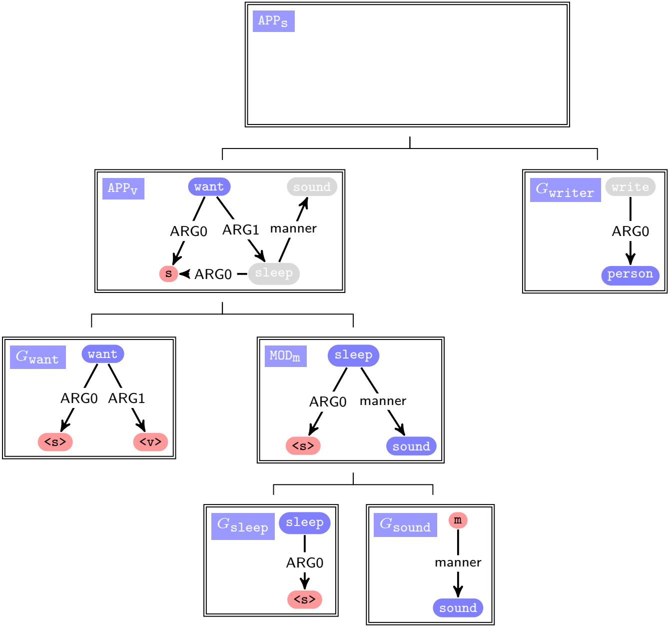
APP_s
APP_v
G_writer
G_want
want
ARG0
ARG1
<s>
<v>
MOD_m
sleep
ARG0
manner
<s>
sound
G_sleep
sleep
ARG0
<s>
G_sound
m
manner
sound

AM Algebra



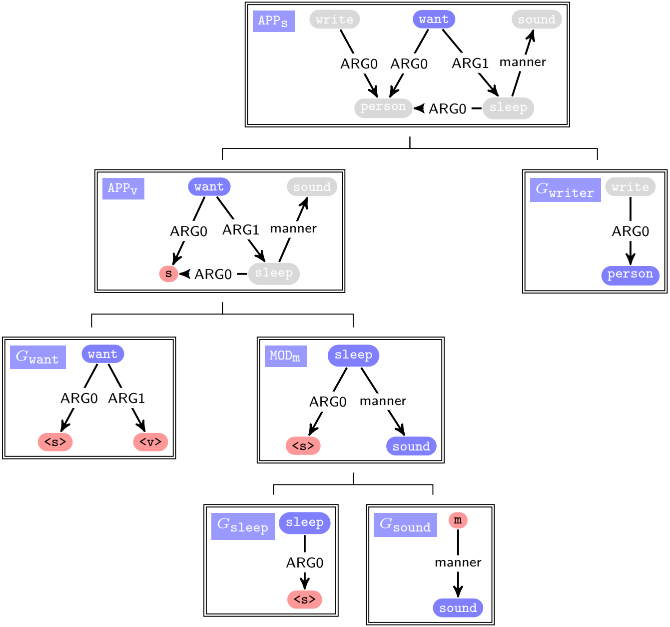
APPs
APPv
want
sound
ARG0
ARG1
manner
s
ARG0
sleep
Gwriter
Gwant
want
ARG0
ARG1
<s>
<v>
MODm
sleep
ARG0
manner
<s>
sound
Gsleep
sleep
ARG0
<s>
Gsound
m
manner
sound

AM Algebra



APPs
APPv
want
sound
ARG0
ARG1
manner
s
ARG0
sleep
Gwriter
write
ARG0
person
Gwant
want
ARG0
ARG1
<s>
<v>
MODm
sleep
ARG0
manner
<s>
sound
Gsleep
sleep
ARG0
<s>
Gsound
m
manner
sound

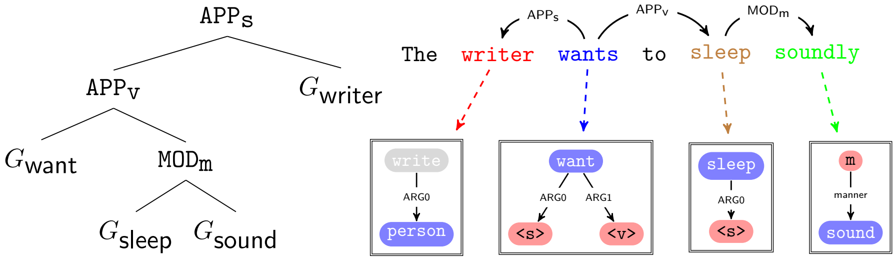
AM Algebra



APP$_s$
write
want
sound
ARG0
ARG0
ARG1
manner
person
ARG0
sleep
APP$_v$
want
sound
ARG0
ARG1
manner
s
ARG0
sleep
$G_{\texttt{writer}}$
write
ARG0
person
$G_{\texttt{want}}$
want
ARG0
ARG1
<s>
<v>
MOD$_m$
sleep
ARG0
manner
<s>
sound
$G_{\texttt{sleep}}$
sleep
ARG0
<s>
$G_{\texttt{sound}}$
m
manner
sound

AM Algebra–Based Parsing

Tree construction + Lexical interpretation ⇒ Meaning representation



- J. Groschwitz, M. Lindemann, M. Fowlie, M. Johnson, and A. Koller. 2018. AMR dependency parsing with a typed semantic algebra.
- M. Lindemann, J. Groschwitz and A. Koller. 2019. Compositional Semantic Parsing Across Graphbanks.

AM Algebra–Based Parsing

Tree construction + Lexical interpretation ⇒ Meaning representation

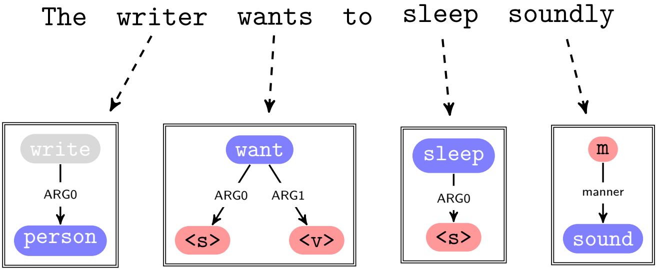
- A supertagger predicts graphs for words.
- A dependency parser predicts `APP` and `MOD` edges.
- At test time, compute highest-scoring well-typed AM dependency tree for input sentence.

`The writer wants to sleep soundly`

AM Algebra–Based Parsing

Tree construction + Lexical interpretation ⇒ Meaning representation

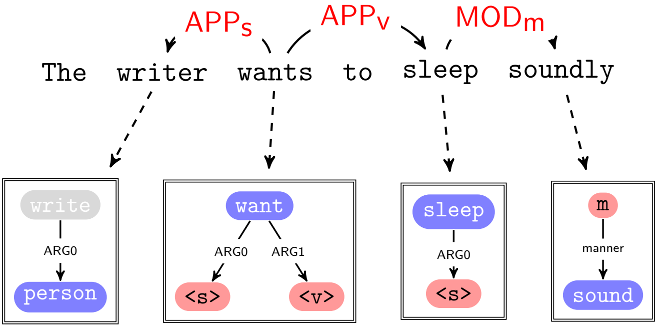
- A supertagger predicts graphs for words.
- A dependency parser predicts `APP` and `MOD` edges.
- At test time, compute highest-scoring well-typed AM dependency tree for input sentence.



AM Algebra–Based Parsing

Tree construction + Lexical interpretation ⇒ Meaning representation

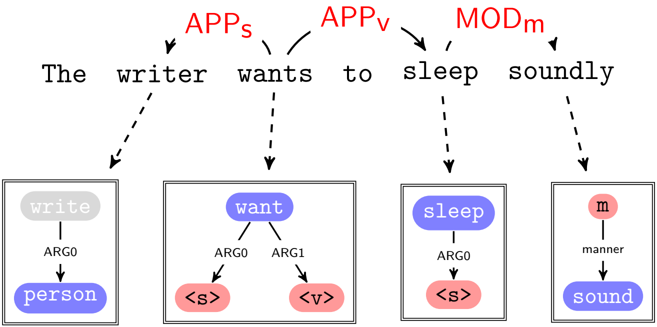
- A supertagger predicts graphs for words.
- A dependency parser predicts APP and MOD edges.
- At test time, compute highest-scoring well-typed AM dependency tree for input sentence.



AM Algebra–Based Parsing

Tree construction + Lexical interpretation ⇒ Meaning representation

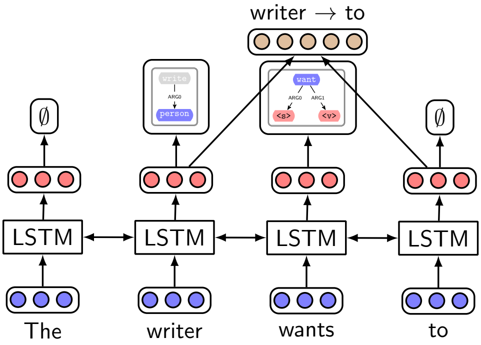
- A supertagger predicts graphs for words.
- A dependency parser predicts APP and MOD edges.
- At test time, compute highest-scoring well-typed AM dependency tree for input sentence.



AM Algebra–Based Parsing

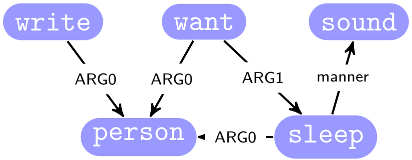
Tree construction + Lexical interpretation ⇒ Meaning representation

- A supertagger predicts graphs for words.
- A dependency parser predicts `APP` and `MOD` edges.
- At test time, compute highest-scoring well-typed AM dependency tree for input sentence.



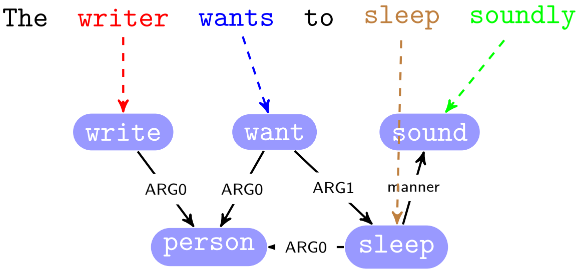
Decomposing the Training Data

The writer wants to sleep soundly



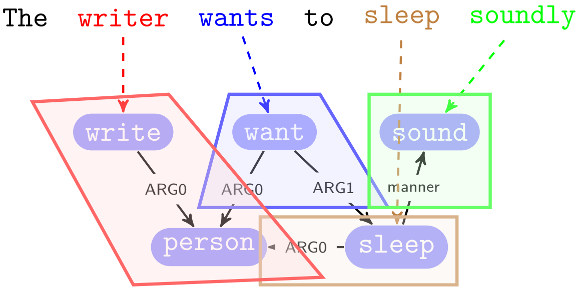
Decompose graph into parts based on concept-to-word alignment.

Decomposing the Training Data



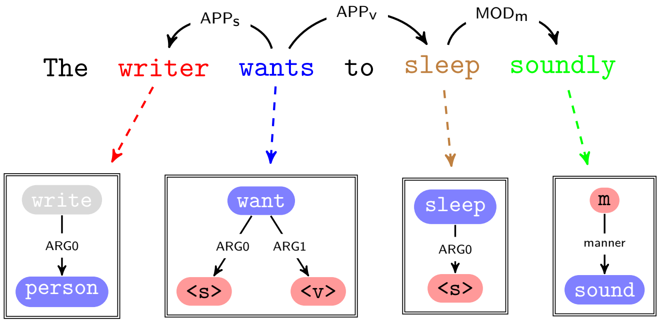
Decompose graph into parts based on concept-to-word alignment.

Decomposing the Training Data



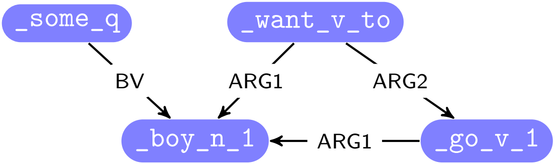
Decompose graph into parts based on concept-to-word alignment.

Decomposing the Training Data



Training data = supertags + AM dependency tree

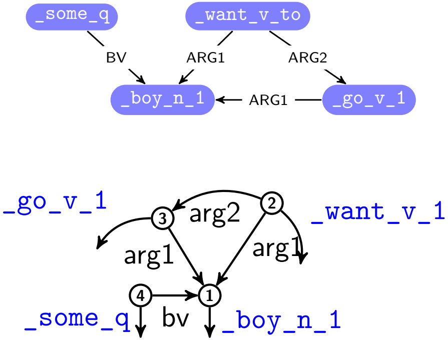
Hypergraph



A graph consists of:

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

Hypergraph



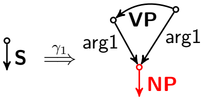
A hypergraph adds:

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

Hyperedge Replacement Grammar

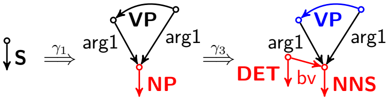
- Terminal vs. non-terminal hyperedges
- Non-terminal hyperedges are utilized to control a derivation process.
- A derivation starts from a non-terminal hyperedge .

Hyperedge Replacement Grammar



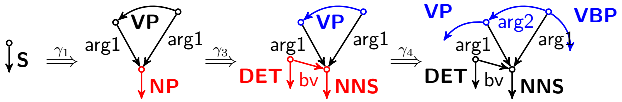
- Terminal vs. non-terminal hyperedges
- Non-terminal hyperedges are utilized to control a derivation process.
- A derivation starts from a non-terminal hyperedge .
- In a derivation step, we substitute a non-terminal hyperedge with a hypergraph .

Hyperedge Replacement Grammar



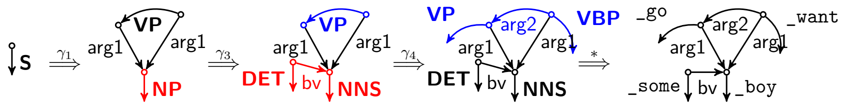
- Terminal vs. non-terminal hyperedges
- Non-terminal hyperedges are utilized to control a derivation process.
- A derivation starts from a non-terminal hyperedge .
- In a derivation step, we substitute a non-terminal hyperedge with a hypergraph .

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 .

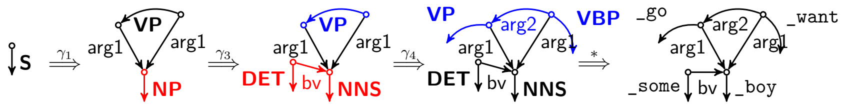
Hyperedge Replacement Grammar



- 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.

Drewes et al. (1997)

Hyperedge Replacement Grammar



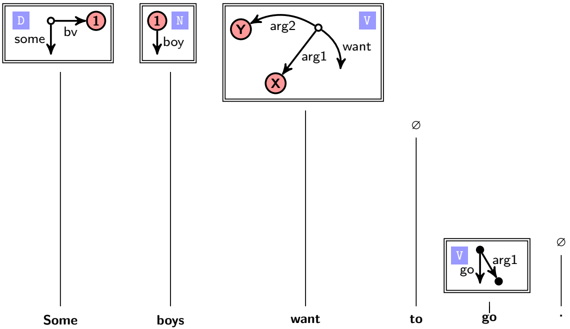
- Terminal vs. non-terminal hyperedges (symbols)
- Non-terminal hyperedges (symbols) are utilized to control a derivation process.
- A derivation starts from a non-terminal hyperedge (symbol).
- In a derivation step, we substitute a non-terminal hyperedge (symbols) with a hypergraph (a sequence of symbols).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

Drewes et al. (1997)

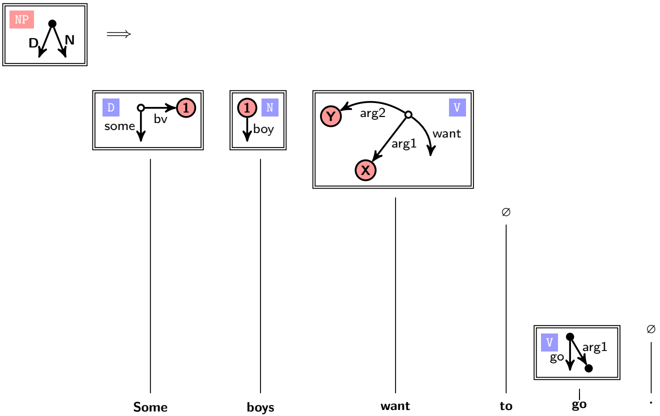
Context-Free Graph Rewriting

Some boys want to go .

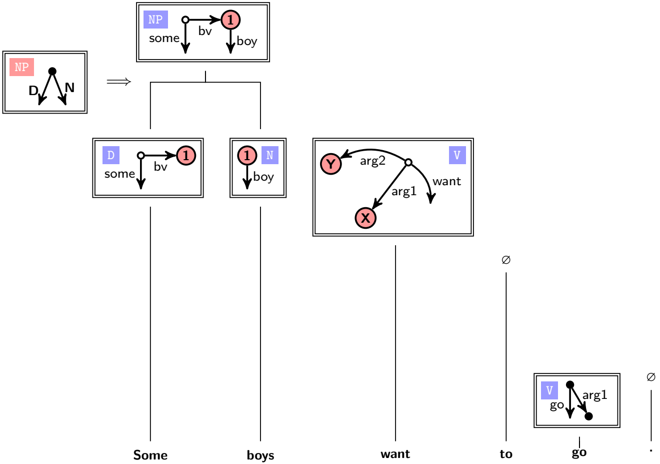
Context-Free Graph Rewriting



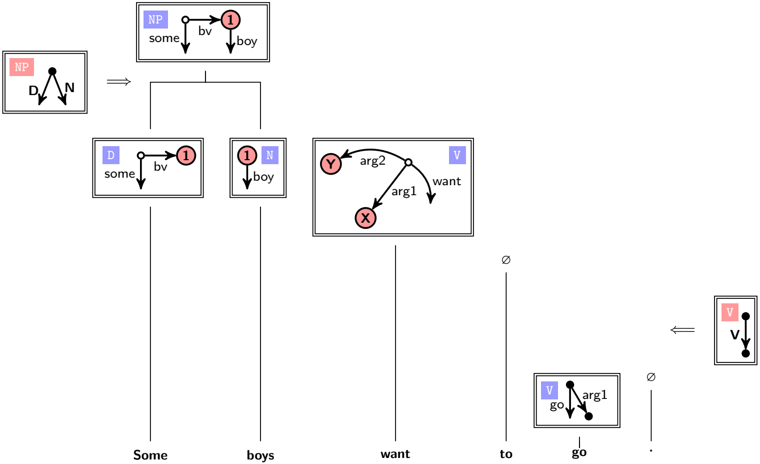
Context-Free Graph Rewriting



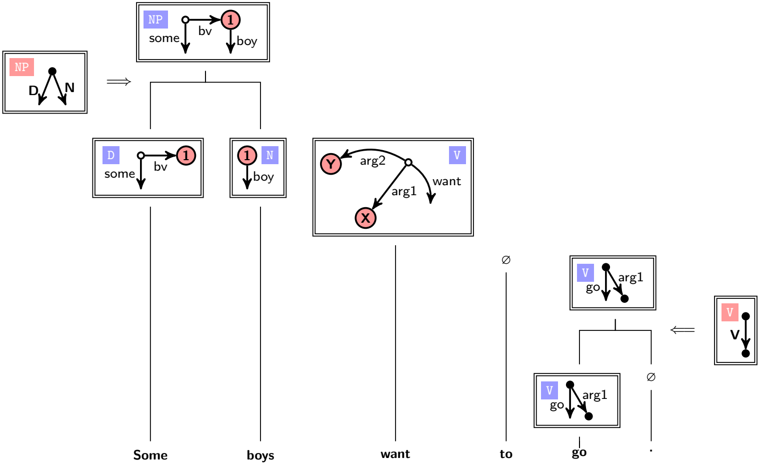
Context-Free Graph Rewriting



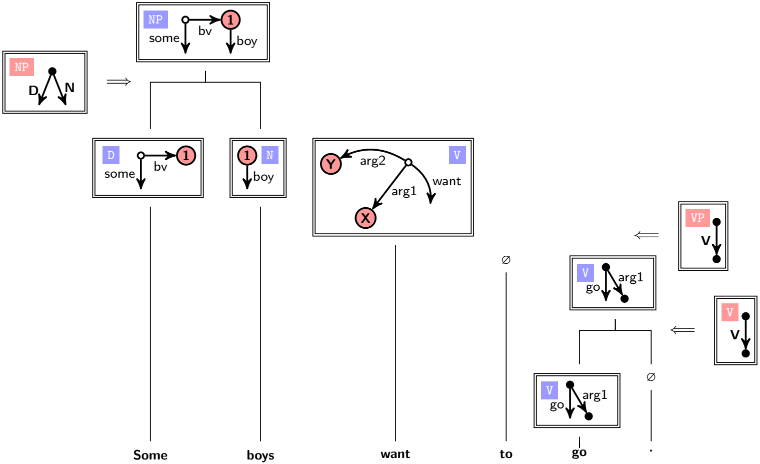
Context-Free Graph Rewriting



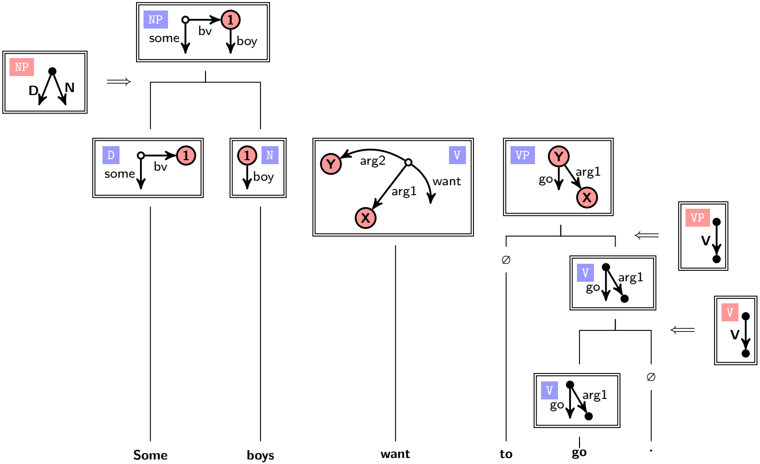
Context-Free Graph Rewriting



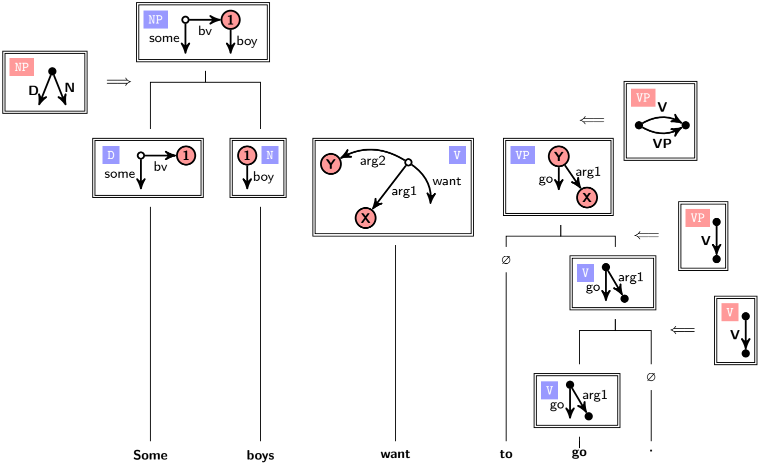
Context-Free Graph Rewriting



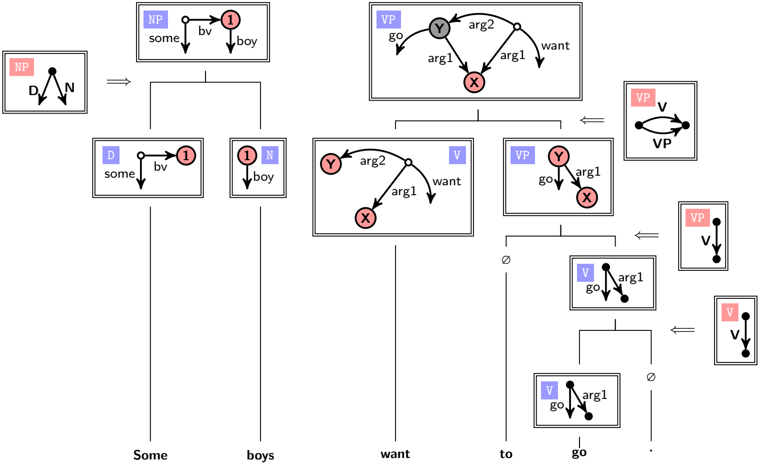
Context-Free Graph Rewriting



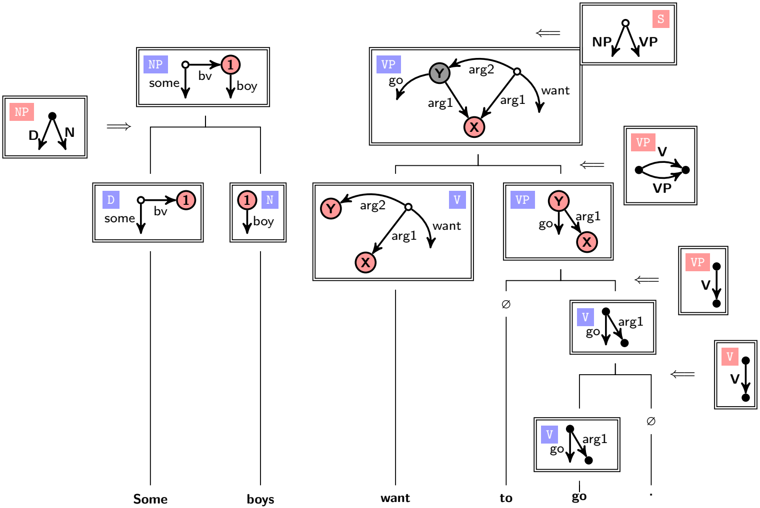
Context-Free Graph Rewriting



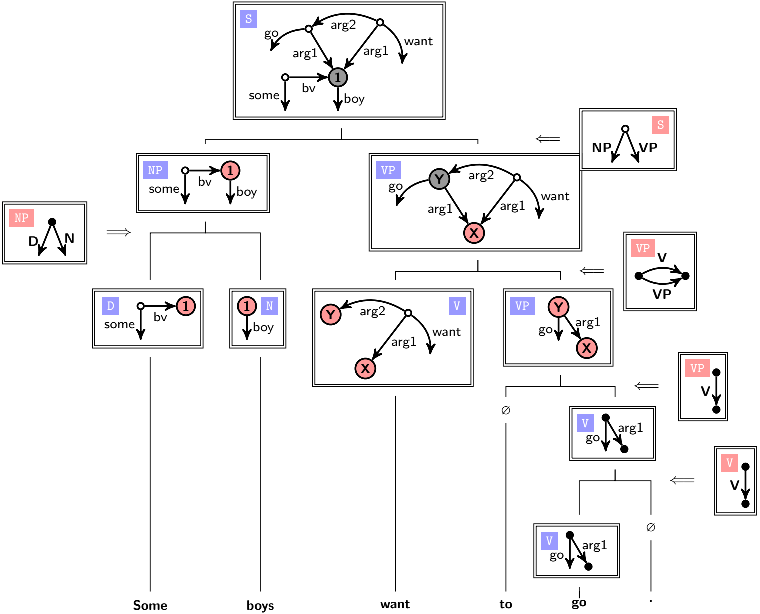
Context-Free Graph Rewriting



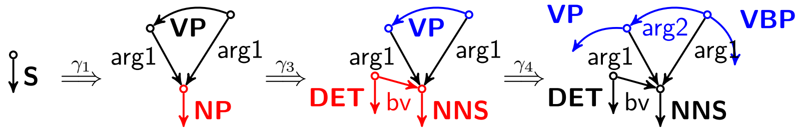
Context-Free Graph Rewriting



Context-Free Graph Rewriting

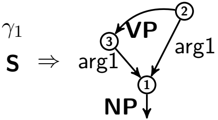


HRGs Can be Linguistically Meaningful



Rules

γ_1



Control construction
(Equi verbs)

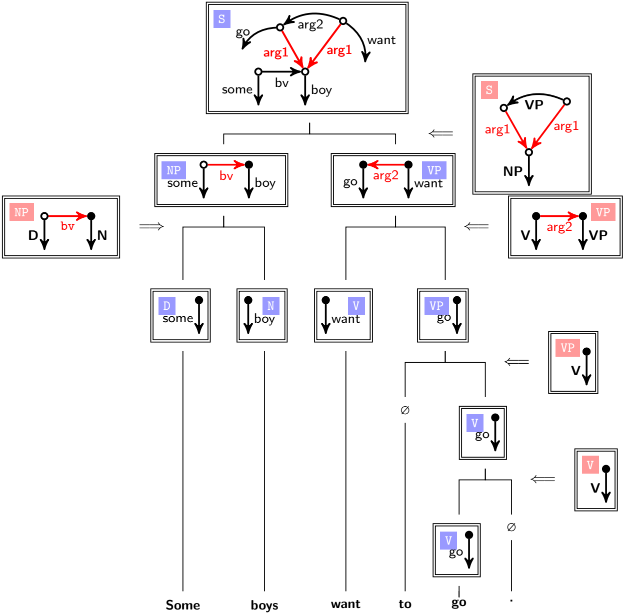
γ_3

Generalized quantifier

γ_4

Predicate–argument structure

Construction Semantics (Revisited)

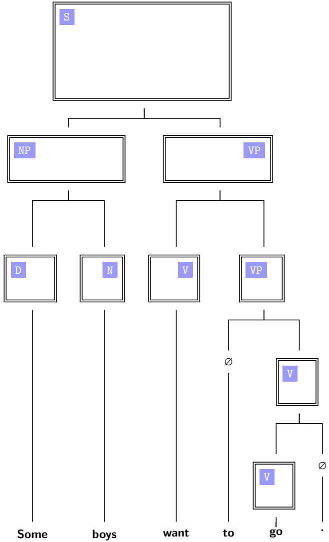


SHRG-Based Parsing

Tree construction $+$ Semantic interpretation $\Rightarrow$ Meaning representation

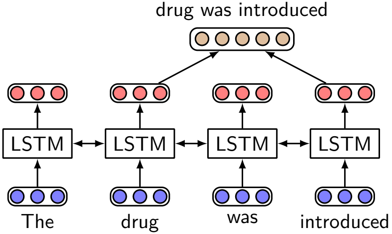
SHRG-Based Parsing

Tree construction + Semantic interpretation ⇒ Meaning representation



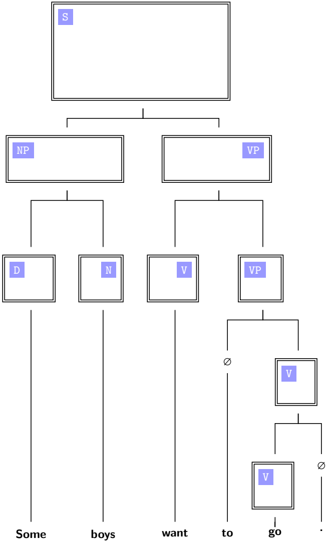
Syntactic parsing

- Word representation: LSTM/Transformer
- Phrase representation: LSTM-minus/Transformer



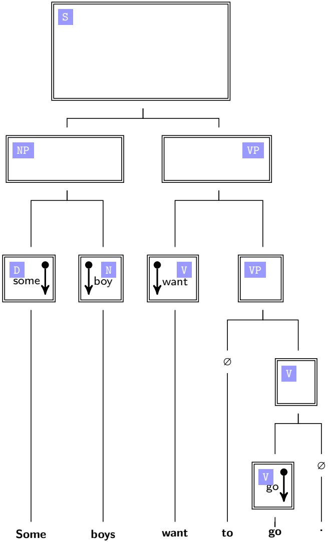
SHRG-Based Parsing

Tree construction + Semantic interpretation ⇒ Meaning representation



SHRG-Based Parsing

Tree construction + Semantic interpretation ⇒ Meaning representation



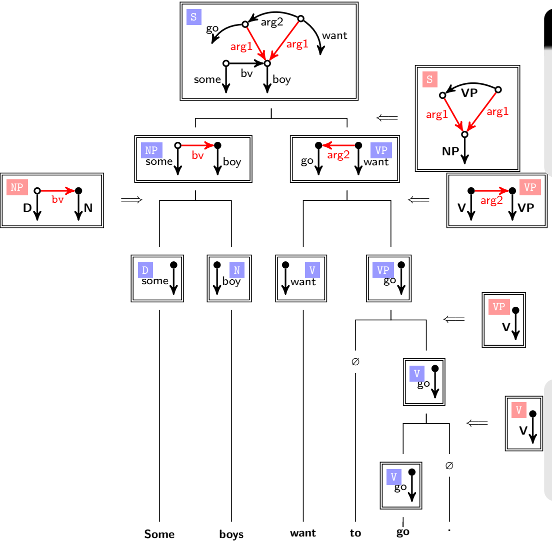
Semantic interpretation

- Rule embedding
- Subgraph embedding
- Beam search decoder

Y. Chen, W. Sun and X. Wan. 2018. Accurate SHRG-based semantic parsing.

SHRG-Based Parsing

Tree construction + Semantic interpretation ⇒ Meaning representation



Semantic interpretation

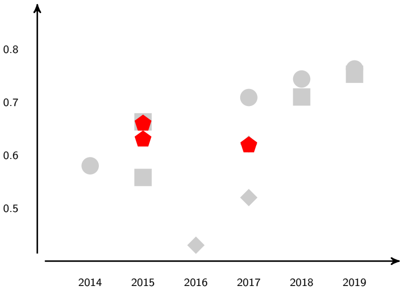
- Rule embedding
- Subgraph embedding
- Beam search decoder

Y. Chen, W. Sun and X. Wan. 2018. Accurate SHRG-based semantic parsing.

Magic Numbers

	DM		PAS		PSD		EDS		AMR	
	id	ood	id	ood	id	ood	Smatch	EDM_{na}	Smatch	Smatch
Groschwitz et al. (2018)	-	-	-	-	-	-	-	-	70.2	71.0
Lyu & Titov (2018)	-	-	-	-	-	-	-	-	73.7	74.4
S. Zhang et al. (2019)	-	-	-	-	-	-	-	-	-	**76.3**
H. Peng et al. (2017)	89.4	84.5	92.2	88.3	77.6	75.3	-	-	-	-
+Multitask learning	90.4	85.3	92.7	89.0	78.5	76.4	-	-	-	-
Dozat & Manning (2018)	93.7	88.9	94.0	90.8	81.0	79.4	-	-	-	-
Buys & Blunsom (2017)	-	-	-	-	-	-	85.5	85.9	60.1	-
Chen, Sun, & Wan (2018)	-	-	-	-	-	-	**90.9**	**90.4**	-	-
Lindemann et al. (2019)	**93.9**	**90.3**	**94.5**	**92.5**	**82.0**	**81.5**	90.1	84.9	**74.3**	75.3
+Multitask learning	**94.1**	**90.5**	**94.7**	**92.8**	**82.1**	**81.6**	90.4	85.2	74.5	75.3

Transition-Based Approaches



- Inspired by the successful design of transition-based dependency parsers.
  - Very early work: Sagae & Tsujii (2008).
- Psycholinguistically motivated: Left-to-right, word-by-word
- Partially parsed results (parsing states) constrain parsing of subsequent words
- Greedy search to get a *good* parse.

Transition-Based Parsing

A transition system for parsing is a quadruple $S = (C, T, c_s, C_t)$, where

1. C is a set of configurations, each of which represents a parser state.
2. T is a set of transitions, each of which represents a parsing action,
3. c_s initializes S by mapping a sentence x to a particular configuration,
4. $C_t \subseteq C$ is a set of terminal configurations.

Deterministic parsing

$\textsc{Parse}(x = (w_0, w_1, \ldots, w_n))$

$c \leftarrow c_s(x)$

while $c \notin C_t$

$\quad c = \textsc{Act}(c, \textsc{GetTransition}(c))$

return G_c

Sagae & Tsujii (2008); Wang et al. (2015c, 2015a); X. Zhang et al. (2016); Buys & Blunsom (2017); Gildea et al. (2018); Sun et al. (2019)

Transition-Based EDS Parsing

Partial analysis

Transition

- Shift (CI)
- Left-Arc (RD)
- Right-Arc (RD)
- Reduce

Stack
[]

Buffer
[Some, boys, want, to, go]

Transition
Init(_some_q) ▷replace Some in the buffer with _some_q

J. Buys and P. Blunsom. 2017. Robust Incremental Neural Semantic Graph Parsing.

Transition-Based EDS Parsing

Partial analysis

_some_q

Transition

- Shift (CI)
- Left-Arc (RD)
- Right-Arc (RD)
- Reduce

Stack
[]

Buffer
[_some_q, boys, want, to, go]

Transition
Init(_some_q) ▷replace Some in the buffer with _some_q

J. Buys and P. Blunsom. 2017. Robust Incremental Neural Semantic Graph Parsing.

Transition-Based EDS Parsing

Partial analysis

_some_q

Transition

- Shift (CI)
- Left-Arc (RD)
- Right-Arc (RD)
- Reduce

Stack
[]

Buffer
[_some_q, boys, want, to, go]

Transition
Shift(_boy_n_1) ▷move _some_q from the buffer to the stack;

J. Buys and P. Blunsom. 2017. Robust Incremental Neural Semantic Graph Parsing.

Transition-Based EDS Parsing

Partial analysis

_some_q

Transition

- Shift (CI)
- Left-Arc (RD)
- Right-Arc (RD)
- Reduce

Stack
[_some_q]

Buffer
[_boy_n_1, want, to, go]

Transition
Shift(_boy_n_1) ▷then boys in the buffer with _boy_n_1

J. Buys and P. Blunsom. 2017. Robust Incremental Neural Semantic Graph Parsing.

Transition-Based EDS Parsing

Partial analysis

_some_q

_boy_n_1

Transition

- Shift (CI)
- Left-Arc (RD)
- Right-Arc (RD)
- Reduce

Stack
[_some_q]

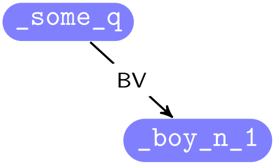
Buffer
[_boy_n_1, want, to, go]

Transition
Right-Arc(BV) ▷link _some_q and _boy_n_1

J. Buys and P. Blunsom. 2017. Robust Incremental Neural Semantic Graph Parsing.

Transition-Based EDS Parsing

Partial analysis



Transition

- Shift (CI)
- Left-Arc (RD)
- Right-Arc (RD)
- Reduce

Stack
[_some_q]

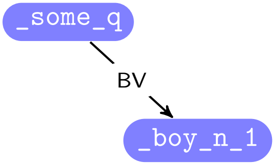
Buffer
[_boy_n_1, want, to, go]

Transition
Right-Arc(BV) ▷link _some_q and _boy_n_1

J. Buys and P. Blunsom. 2017. Robust Incremental Neural Semantic Graph Parsing.

Transition-Based EDS Parsing

Partial analysis



Transition

- Shift (CI)
- Left-Arc (RD)
- Right-Arc (RD)
- Reduce

Stack
[_some_q]

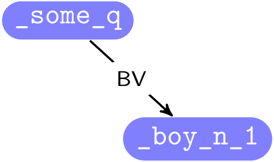
Buffer
[_boy_n_1, want, to, go]

Transition
Reduce ▷remove _some_q from the stack

J. Buys and P. Blunsom. 2017. Robust Incremental Neural Semantic Graph Parsing.

Transition-Based EDS Parsing

Partial analysis



Transition

- Shift (CI)
- Left-Arc (RD)
- Right-Arc (RD)
- Reduce

Stack
[]

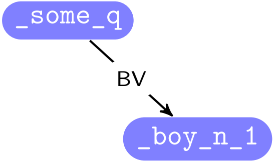
Buffer
[_boy_n_1, want, to, go]

Transition
Reduce ▷remove _some_q from the stack

J. Buys and P. Blunsom. 2017. Robust Incremental Neural Semantic Graph Parsing.

Transition-Based EDS Parsing

Partial analysis



Transition

- Shift (CI)
- Left-Arc (RD)
- Right-Arc (RD)
- Reduce

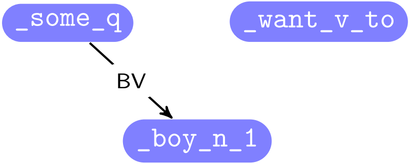
Stack
[]

Buffer
[_boy_n_1, want, to, go]

Transition
Shift(_want_v_to) ▷move _boy_n_1 from the buffer to stack;

J. Buys and P. Blunsom. 2017. Robust Incremental Neural Semantic Graph Parsing.

Partial analysis



Transition

- Shift (CI)
- Left-Arc (RD)
- Right-Arc (RD)
- Reduce

Stack
[_boy_n_1]

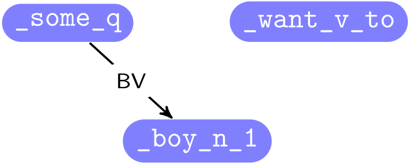
Buffer
[_want_v_to, to, go]

Transition
Shift(_want_v_to) ▷then replace want in the buffer with _want_v_to

J. Buys and P. Blunsom. 2017. Robust Incremental Neural Semantic Graph Parsing.

Transition-Based EDS Parsing

Partial analysis



Transition

- Shift (CI)
- Left-Arc (RD)
- Right-Arc (RD)
- Reduce

Stack
[_boy_n_1]

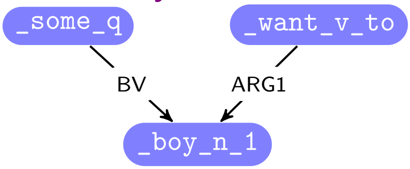
Buffer
[_want_v_to, to, go]

Transition
Left-Arc(ARG1) ▷link _boy_n_1 and _want_v_to

J. Buys and P. Blunsom. 2017. Robust Incremental Neural Semantic Graph Parsing.

Transition-Based EDS Parsing

Partial analysis



Transition

- Shift (CI)
- Left-Arc (RD)
- Right-Arc (RD)
- Reduce

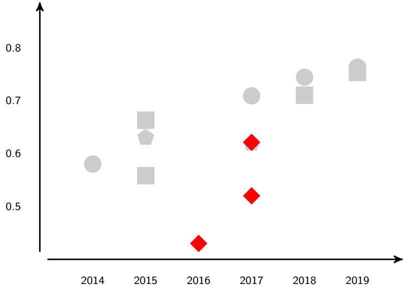
Stack
[_boy_n_1]

Buffer
[_want_v_to, to, go]

Transition
Left-Arc(ARG1) ▷link _boy_n_1 and _want_v_to

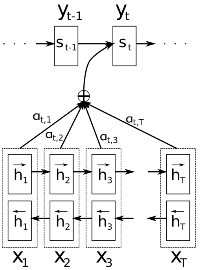
J. Buys and P. Blunsom. 2017. Robust Incremental Neural Semantic Graph Parsing.

Translation-Based Approaches

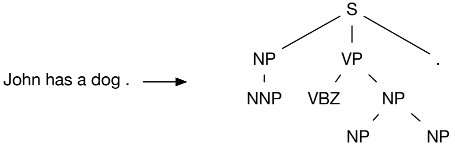


- Inspired by neural machine translation.
- Semantic graph as a foreign language.
- A parser knows how to linearize a graph.
- Data augmentation has been shown very helpful, partially reflecting the *data-hungry nature* of `seq2seq` models.

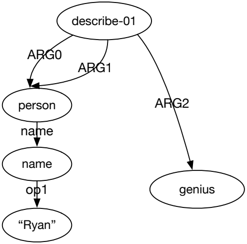
Sequence-to-Sequence Models for Parsing



Tree as a foreign language



John has a dog . ⟶ (S (NP NNP)NP (VP VBZ (NP DT NN)NP)VP .)S



Graph as a foreign language

```
-TOP-( describe-01 ARG0( person
name( name op1( "Ryan" )op1 )name
)ARG0 ARG1( person -RET- )ARG1
ARG2( genius )ARG2 )-TOP-
```

Many variants

X. Peng et al. (2017); Konstas et al. (2017); Wang et al. (2015b, 2015a); Buys & Blunsom (2017)

Integrated with Other Approaches

Translation + Transition–Based Approach

- Using a sequence-to-sequence model
- The input sequence is the sequence of input words
- The output sequence is a sequence of transitions that leads to an output graph

X. Peng et al. (2018)

Translation + Factorization–Based Approach

- Identifying concepts with a sequence-to-sequence model
- Linking nodes with a factorization model

S. Zhang et al. (2018)

Cross-Framework Parsing

- State-of-the-art graph parsers rely on large-scale, manually annotated graphbanks.
- Multiple heterogeneous graphbanks EXIST!

$\xrightarrow{?}$ Can we get a better parser by integrating heterogeneous graphbanks?

1. Heterogeneous annotations are (similar but) *different*.
   - Different projects, different linguistic theories, different representation formalisms, different annotation schemes, etc.

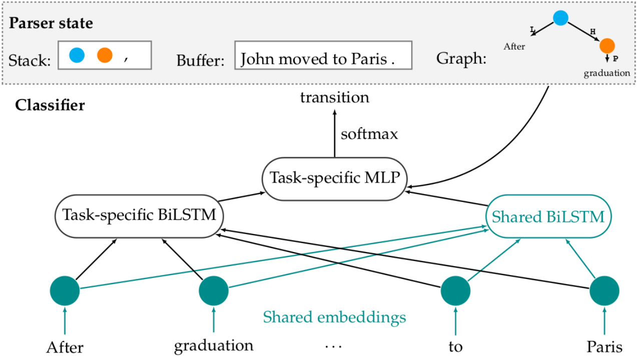
   $\rightarrow$ Reducing approximation errors
2. Heterogeneous annotations are (different but) *similar*.
   - Similar high-level linguistic principles.

   $\rightarrow$ Reducing estimation errors

Multitask Learning for Semantic Parsing

Freda (Daumé III, 2007)

Hal Daumé III. 2007. Frustratingly easy domain adaptation.



D. Hershcovich, O. Abend and A. Rappoport. 2018. Multitask Parsing Across Semantic Representations

Magic Numbers

Improving a Transition-Based Parser for UCCA (Hershcovich et al., 2018)

	In-Domain		**Out-of-Domain**	
	Primary	Romote	Primary	Romote
Single	73.6	51.5	69	26.7
+AMR	73.7	49.9	69.5	27.5
+DM	74.8	53.9	70.7	25.9
+UD	74.1	50.8	69.7	28.7

Improving a Factorization-Based Parser for Many Graphbanks (Lindemann et al., 2019)

	DM		**PAS**		**PSD**		**EDS**	**AMR**
	id	ood	id	ood	id	ood	$\mathrm{EDM}_{\mathrm{na}}$	Smatch
Single (GloVe)	90.4	84.3	91.4	86.6	78.1	74.5	82.5	69.2
+MTL	91.2	85.7	92.2	88.0	78.9	76.2	83.3	70.4
Single (BERT)	93.9	90.3	94.5	92.5	82.0	81.5	84.9	74.3
+MTL	94.1	90.5	94.7	92.8	82.1	81.6	85.2	74.5

Magic Numbers

Improving a Transition-Based Parser for UCCA (Hershcovich et al., 2018)

	In-Domain		**Out-of-Domain**	
	Primary	Romote	Primary	Romote
Single	73.6	51.5	69	26.7
+AMR	73.7	49.9	69.5	27.5
+DM	74.8	53.9	70.7	25.9
+UD	74.1	50.8	69.7	28.7

Improving a Factorization-Based Parser for Many Graphbanks (Lindemann et al., 2019)

	DM		**PAS**		**PSD**		**EDS**	**AMR**
	id	ood	id	ood	id	ood	EDM_{na}	Smatch
Single (GloVe)	90.4	84.3	91.4	86.6	78.1	74.5	82.5	69.2
+MTL	91.2	85.7	92.2	88.0	78.9	76.2	83.3	70.4
Single (BERT)	93.9	90.3	94.5	92.5	82.0	81.5	84.9	74.3
+MTL	94.1	90.5	94.7	92.8	82.1	81.6	85.2	74.5

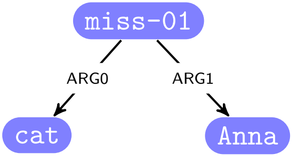
Cross-lingual parsing

Multilingual Parsing

- SemEval 2016: Chinese Semantic Dependency Parsing
- SemEval 2019: Cross-lingual semantic parsing with UCCA
  - English, German, French
- One single parsing architecture for many languages

Cross-lingual parsing

- Mapping a string of $\mathcal{L}_A$ to a graph of $\mathcal{L}_B$
- Motivation: Don't create a graphbank for $\mathcal{L}_A$.



EN: *Anna's cat is missing her*
DE: *Anna fehlt ihrem Kater*

5 Using Semantic Structure

Applications of Semantic Graphs

- Using semantic graphs in applications may improve accuracy:
  - Semantic graphs abstract over surface variation.
  - Easier to generalize over graphs than over sentences.
  - ... if semantic parsing is accurate enough.

- Typical applications:
  - machine translation Jones et al. (2012)
  - entity linking / KB population Reddy et al. (2014); Pan et al. (2015)
  - summarization Liu et al. (2015); Hardy & Vlachos (2018)

- Original motivation for AMR: semantics-based SMT.
- Initial work of Jones et al. (2012) yielded results that were promising at the time.
- Steamrolled by end-to-end neural methods for MT.

Example

- Original motivation for AMR: semantics-based SMT.
- Initial work of Jones et al. (2012) yielded results that were promising at the time.
- Steamrolled by end-to-end neural methods for MT.

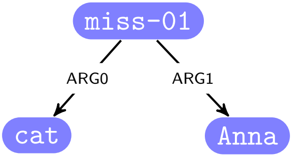
Example

DE: *Anna fehlt ihrem Kater*

- Original motivation for AMR: semantics-based SMT.
- Initial work of Jones et al. (2012) yielded results that were promising at the time.
- Steamrolled by end-to-end neural methods for MT.

Example

DE: *Anna fehlt ihrem Kater* $\longrightarrow$

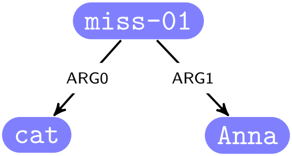


Machine Translation

- Original motivation for AMR: semantics-based SMT.
- Initial work of Jones et al. (2012) yielded results that were promising at the time.
- Steamrolled by end-to-end neural methods for MT.

Example

DE: *Anna fehlt ihrem Kater* ⟶



EN: *Anna's cat is missing her* ⟵

Entity Linking

Task: Map entities in text to entities in a knowledge base (KB).

Entity Linking

Task: Map entities in text to entities in a knowledge base (KB).

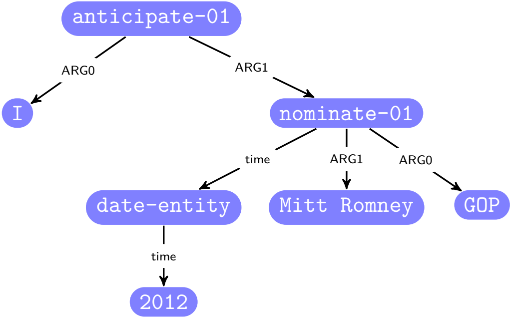
I am cautiously anticipating the GOP nominee in 2012 not to be Mitt Romney.

Pan et al. (2015)

Entity Linking

Task: Map entities in text to entities in a knowledge base (KB).

I am cautiously anticipating the GOP nominee in 2012 not to be Mitt Romney.



Pan et al. (2015)

Entity Linking

Task: Map entities in text to entities in a knowledge base (KB).

I am cautiously anticipating the GOP nominee in 2012 not to be Mitt Romney.

Romney was the Governor of Massachusetts ...

Romney is the great-great-grandson of a Mormon pioneer

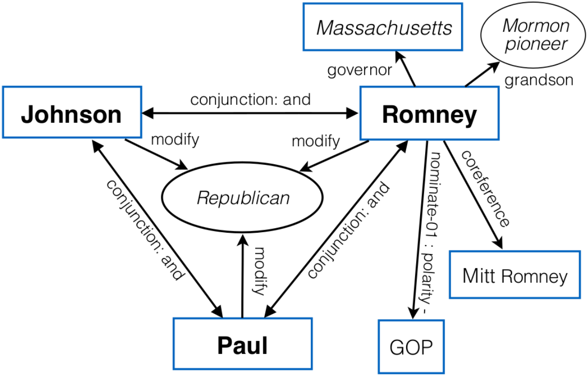
Republican candidates like Romney, Paul, and Johnson ...

Pan et al. (2015)

Entity Linking

Task: Map entities in text to entities in a knowledge base (KB).

I am cautiously anticipating the GOP nominee in 2012 not to be Mitt Romney.
Romney was the Governor of Massachusetts ...
Romney is the great-great-grandson of a Mormon pioneer
Republican candidates like Romney, Paul, and Johnson ...

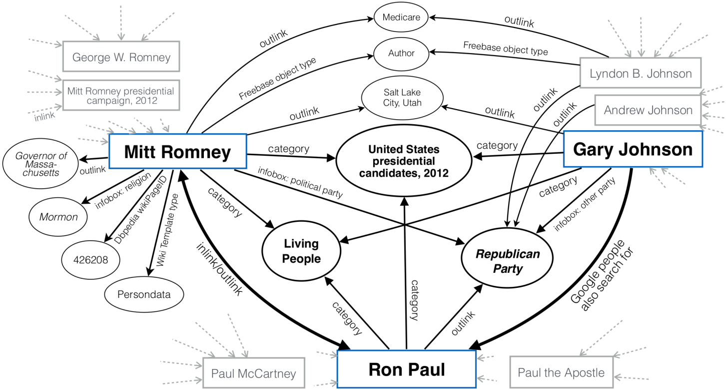


Pan et al. (2015)

Entity Linking

Task: Map entities in text to entities in a knowledge base (KB).

I am cautiously anticipating the GOP nominee in 2012 not to be Mitt Romney.
Romney was the Governor of Massachusetts ...
Romney is the great-great-grandson of a Mormon pioneer
Republican candidates like Romney, Paul, and Johnson ...



Pan et al. (2015)

Entity Linking: Evaluation

Evaluation on corpus with 1600 entity mentions, manually annotated with AMRs and entity links.

approach	news	forums	total
non-graph SOTA	93.1	87.4	91.0
"collaborator", human AMR	93.7	86.9	91.1
"collaborator", JAMR	90.2	85.7	88.5
"combined", human AMR	94.3	88.3	92.1

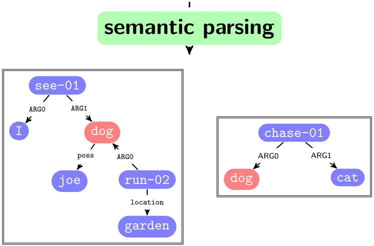
Pan et al. (2015)

I saw Joe's dog, which was running in the garden.
The dog was chasing a cat.

Liu et al. (2015); Hardy & Vlachos (2018)

Abstractive Summarization

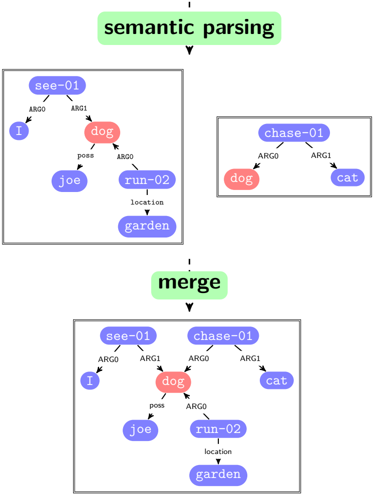
I saw Joe's dog, which was running in the garden.
The dog was chasing a cat.



Liu et al. (2015); Hardy & Vlachos (2018)

Abstractive Summarization

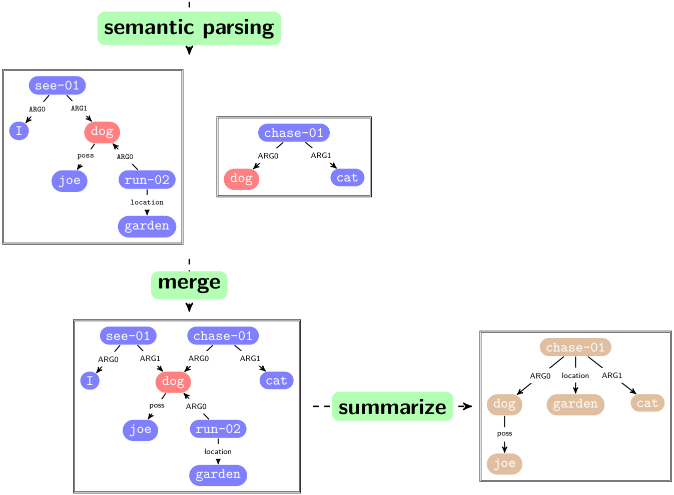
I saw Joe's dog, which was running in the garden.
The dog was chasing a cat.



Liu et al. (2015); Hardy & Vlachos (2018)

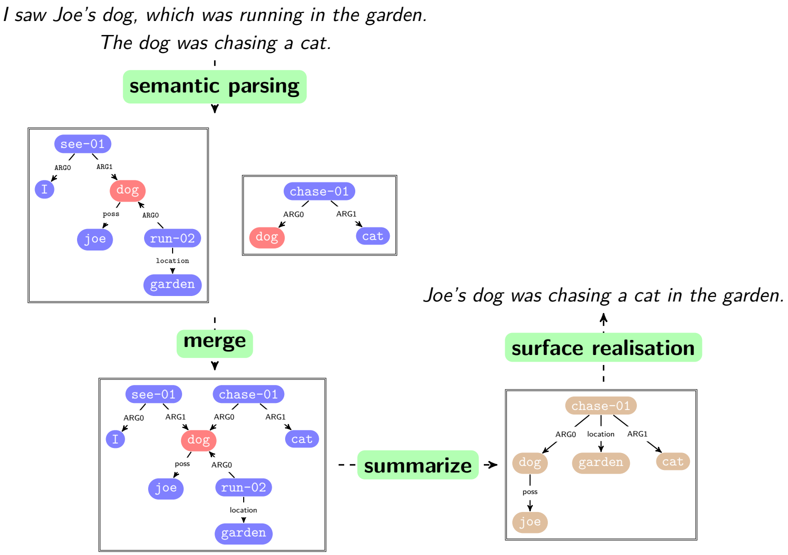
Abstractive Summarization

I saw Joe's dog, which was running in the garden.
The dog was chasing a cat.



Liu et al. (2015); Hardy & Vlachos (2018)

Abstractive Summarization



Liu et al. (2015); Hardy & Vlachos (2018)

Evaluation on Proxy Report section of AMRBank LCD2017T10.

AMRs	NLG model	Rouge-1	Rouge-2	Rouge-L
gold	amr2seq + LM	40.4	20.3	31.4
	amr2seq	38.9	12.9	27.0
	amr2bow (Liu et al.)	39.6	6.2	22.1
RIGA	amr2seq + LM	42.3	21.2	33.6
	amr2seq	37.8	10.7	26.9
–	OpenNMT	36.1	19.2	31.1

Hardy & Vlachos (2018)

6

Conclusions
Outlook

Conclusion

Semantic graph parsing: a success story

- Capture semantic information that is not explicit in syntactic parses.
- Parsers getting increasingly accurate.
- Graphs seem useful in applications.
- Look out for graph parsing papers throughout ACL 2019.

Differences between graphbanks are substantial

- Anchoring of nodes in tokens (flavors 0–2).
- Capture different facets of meaning.
- Different design choices.

Outlook

Cross-Framework Semantic Parsing

- Most graph parsers work only for one flavor of graphbank.
- Generalize across graphbanks?
- Check out CoNLL 2019 Shared Task on Cross-Framework Meaning Representation Parsing (MRP, `http://mrp.nlpl.eu/`).

Facets of Meaning

- Many facets of meaning are not represented by graphbanks.
- What facets are relevant for what applications?
- Push graphbanks so they can be represented, or switch to different meaning representations.
- Check out the ACL 2019 Workshop on Designing Meaning Representations (`https://www.cs.brandeis.edu/~clp/dmr/`).

Acknowledgments

With many thanks to Dan Flickinger, Emily Bender, Ann Copestake, and the DELPH-IN community at large; *the Chinese student who helped with graph preparation*; Noortje Venhuizen *and others* for comments on specific meaning banks.

All mistakes and oversights are, of course, our own.

Questions?

References I

Abend, O., & Rappoport, A. (2013). Universal Conceptual Cognitive Annotation (UCCA). In *Proceedings of the 51th Meeting of the Association for Computational Linguistics* (pp. 228–238). Sofia, Bulgaria.

Abzianidze, L., Bjerva, J., Evang, K., Haagsma, H., van Noord, R., Ludmann, P., . . . Bos, J. (2017). The Parallel Meaning Bank. Towards a multilingual corpus of translations annotated with compositional meaning representations. In *Proceedings of the 15th Meeting of the European Chapter of the Association for Computational Linguistics.* Valencia, Spain.

Artzi, Y., Lee, K., & Zettlemoyer, L. (2015). Broad-coverage CCG semantic parsing with AMR. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing* (pp. 1699–1710). Lisbon, Portugal.

References II

Baker, C. F., Fillmore, C. J., & Lowe, J. B. (1998). The Berkeley FrameNet project. In *Proceedings of the 17th International Conference on Computational Linguistics and the 36th Meeting of the Association for Computational Linguistics* (pp. 86–90). Stroudsburg, PA, USA.

Banarescu, L., Bonial, C., Cai, S., Georgescu, M., Griffitt, K., Hermjakob, U., ... Schneider, N. (2013). Abstract Meaning Representation for sembanking. In *Proceedings of the 7th Linguistic Annotation Workshop and Interoperability with Discourse* (pp. 178–186). Sofia, Bulgaria.

Barzdins, G., & Gosko, D. (2016). RIGA at SemEval-2016 task 8: Impact of Smatch extensions and character-level neural translation on AMR parsing accuracy. In *Proceedings of the 10th international workshop on semantic evaluation (SemEval-2016)* (pp. 1143–1147). San Diego, California.

References III

Basile, V., Bos, J., Evang, K., & Venhuizen, N. (2012). Developing a large semantically annotated corpus. In *Proceedings of the 8th International Conference on Language Resources and Evaluation* (pp. 3196–3200). Istanbul, Turkey.

Bender, E. M., Flickinger, D., Oepen, S., Packard, W., & Copestake, A. (2015). Layers of interpretation. On grammar and compositionality. In *Proceedings of the 11th International Conference on Computational Semantics* (pp. 239–249). London, UK.

Böhmová, A., Hajič, J., Hajičová, E., & Hladká, B. (2003). The Prague Dependency Treebank: A three-level annotation scenario. In A. Abeillé (Ed.), *Treebanks. Building and using parsed corpora* (pp. 103–127). Dordrecht, The Netherlands: Kluwer Academic Publishers.

Buys, J., & Blunsom, P. (2017). Robust incremental neural semantic graph parsing. In *Proceedings of the 55th Meeting of the Association for Computational Linguistics* (pp. 158–167). Vancouver, Canada.

References IV

Cao, J., Huang, S., Sun, W., & Wan, X. (2017a). Parsing to 1-endpoint-crossing, pagenumber-2 graphs. In *Proceedings of the 55th Meeting of the Association for Computational Linguistics* (pp. 2110–2120). Vancouver, Canada.

Cao, J., Huang, S., Sun, W., & Wan, X. (2017b). Quasi-second-order parsing for 1-endpoint-crossing, pagenumber-2 graphs. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing* (pp. 24–34). Copenhagen, Denmark.

Carter, D. (1997). The TreeBanker. A tool for supervised training of parsed corpora. In *Proceedings of the Workshop on Computational Environments for Grammar Development and Linguistic Engineering* (pp. 9–15). Madrid, Spain.

Chen, Y., Huang, S., Wang, F., Cao, J., Sun, W., & Wan, X. (2018). Neural maximum subgraph parsing for cross-domain semantic dependency analysis. In *Proceedings of the 22nd Conference on Natural Language Learning* (pp. 562–572). Brussels, Belgium.

References V

Chen, Y., Sun, W., & Wan, X. (2018). Accurate SHRG-based semantic parsing. In *Proceedings of the 56th Meeting of the Association for Computational Linguistics* (pp. 408–418). Melbourne, Australia.

Copestake, A. (2009). Slacker semantics. Why superficiality, dependency and avoidance of commitment can be the right way to go. In *Proceedings of the 12th Meeting of the European Chapter of the Association for Computational Linguistics* (pp. 1–9). Athens, Greece.

Copestake, A., Flickinger, D., Pollard, C., & Sag, I. A. (2005). Minimal Recursion Semantics. An introduction. *Research on Language and Computation*, *3*(4), 281–332.

Damonte, M., & Cohen, S. B. (2018). Cross-lingual Abstract Meaning Representation parsing. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics* (pp. 1146–1155). New Orleans, LA, USA.

Daumé III, H. (2007). Frustratingly easy domain adaptation. In *Proceedings of the 45th Meeting of the Association for Computational Linguistics* (pp. 256–263). Prague, Czech Republic.

Dozat, T., & Manning, C. D. (2018). Simpler but more accurate semantic dependency parsing. In *Proceedings of the 56th Meeting of the Association for Computational Linguistics* (p. fromto484490). Melbourne, Australia.

Drewes, F., Kreowski, H.-J., & Habel, A. (1997). Hyperedge Replacement Graph Grammars. In G. Rozenberg (Ed.), *Handbook of graph grammars and computing by graph transformation* (pp. 95–162). River Edge, NJ, USA: World Scientific Publishing Co., Inc.

Du, Y., Zhang, F., Zhang, X., Sun, W., & Wan, X. (2015). Peking: Building semantic dependency graphs with a hybrid parser. In *Proceedings of the 9th International Workshop on Semantic Evaluation* (pp. 927–931). Denver, CO, USA.

References VII

Flanigan, J., Thomson, S., Carbonell, J., Dyer, C., & Smith, N. A. (2014). A discriminative graph-based parser for the Abstract Meaning Representation. In *Proceedings of the 52nd Meeting of the Association for Computational Linguistics* (pp. 1426–1436). Baltimore, MD, USA.

Flickinger, D. (2000). On building a more efficient grammar by exploiting types. *Natural Language Engineering*, *6 (1)*, 15–28.

Flickinger, D., Oepen, S., & Bender, E. M. (2017). Sustainable development and refinement of complex linguistic annotations at scale. In N. Ide & J. Pustejovsky (Eds.), *Handbook of linguistic annotation* (pp. 353–377). Dordrecht, The Netherlands: Springer.

Foland, W., & Martin, J. H. (2017). Abstract meaning representation parsing using LSTM recurrent neural networks. In *Proceedings of the 55th Meeting of the Association for Computational Linguistics* (pp. 463–472). Vancouver, Canada: Association for Computational Linguistics.

Gildea, D., Satta, G., & Peng, X. (2018). Cache transition systems for graph parsing. *Computational Linguistics*, *44*(1), 85–118. doi: 10.1162/COLI_a_00308

Groschwitz, J., Fowlie, M., Johnson, M., & Koller, A. (2017). A constrained graph algebra for semantic parsing with AMRs. In *IWCS 2017 - 12th international conference on computational semantics - long papers.*

Groschwitz, J., Lindemann, M., Fowlie, M., Johnson, M., & Koller, A. (2018). AMR dependency parsing with a typed semantic algebra. In *Proceedings of the 56th Meeting of the Association for Computational Linguistics* (pp. 1831–1841). Melbourne, Australia.

Hajič, J., Bejček, E., Bémová, A., Buráňová, E., Hajičová, E., Havelka, J., ... Žabokrtský, Z. (2018). *Prague dependency treebank 3.5.* (LINDAT/CLARIN digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University)

Hajič, J., Hajičová, E., Panevová, J., Sgall, P., Bojar, O., Cinková, S., . . . Žabokrtský, Z. (2012). Announcing Prague Czech-English Dependency Treebank 2.0. In *Proceedings of the 8th International Conference on Language Resources and Evaluation* (pp. 3153–3160). Istanbul, Turkey.

Hardy, & Vlachos, A. (2018). Guided neural language generation for abstractive summarization using abstract meaning representation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing.* Brussels, Belgium.

Hershcovich, D., Abend, O., & Rappoport, A. (2018). Multitask parsing across semantic representations. In *Proceedings of the 56th Meeting of the Association for Computational Linguistics* (pp. 373–385). Melbourne, Australia.

Hockenmaier, J., & Steedman, M. (2007). CCGbank. A corpus of CCG derivations and dependency structures extracted from the Penn Treebank. *Computational Linguistics*, *33*, 355–396.

References X

Ivanova, A., Oepen, S., Øvrelid, L., & Flickinger, D. (2012). Who did what to whom? A contrastive study of syntacto-semantic dependencies. In *Proceedings of the 6th Linguistic Annotation Workshop* (pp. 2–11). Jeju, Republic of Korea.

Jones, B., Andreas, J., Bauer, D., Hermann, K. M., & Knight, K. (2012). Semantics-based machine translation with hyperedge replacement grammars. In *Proceedings of the 24th International Conference on Computational Linguistics.* Mumbai, India.

Konstas, I., Iyer, S., Yatskar, M., Choi, Y., & Zettlemoyer, L. (2017). Neural AMR. Sequence-to-sequence models for parsing and generation. In *Proceedings of the 55th Meeting of the Association for Computational Linguistics* (pp. 146–157). Vancouver, Canada.

Kuhlmann, M., & Jonsson, P. (2015). Parsing to noncrossing dependency graphs. *Transactions of the Association for Computational Linguistics*, *3*, 559–570.

Kuhlmann, M., & Oepen, S. (2016). Towards a catalogue of linguistic graph banks. *Computational Linguistics*, *42*(4), 819–827.

Li, B., Wen, Y., Weiguang, Q. U., Bu, L., & Xue, N. (2016). Annotating the Little Prince with Chinese AMRs. In *Proceedings of the linguistic annotation workshop.*

Lindemann, M., Groschwitz, J., & Koller, A. (2019). Compositional semantic parsing across graphbanks. In *Proceedings of the 57th Meeting of the Association for Computational Linguistics.* Florence, Italy.

Liu, F., Flanigan, J., Thomson, S., Sadeh, N., & Smith, N. A. (2015). Toward abstractive summarization using semantic representations. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics.* Denver, CO, USA.

Lyu, C., & Titov, I. (2018). AMR parsing as graph prediction with latent alignment. In *Proceedings of the 56th Meeting of the Association for Computational Linguistics* (p. fromto397407). Melbourne, Australia.

References XII

McDonald, R., & Pereira, F. (2006). Online learning of approximate dependency parsing algorithms. In *Proceedings of the 11th Meeting of the European Chapter of the Association for Computational Linguistics* (pp. 81–88). Trento, Italy.

Miyao, Y. (2006). *From linguistic theory to syntactic analysis. Corpus-oriented grammar development and feature forest model*. Doctoral dissertation, University of Tokyo, Tokyo, Japan.

Nerbonne, J. (1994). Book review. Computational linguistics and formal semantics. *Computational Linguistics*, *20*(1), 131–136.

Oepen, S., Flickinger, D., Toutanova, K., & Manning, C. D. (2002). Lingo Redwoods. A rich and dynamic treebank for HPSG. In *Proceedings of the 1st International Workshop on Treebanks and Linguistic Theories* (pp. 139–149). Sozopol, Bulgaria.

Oepen, S., Flickinger, D., Toutanova, K., & Manning, C. D. (2004). LinGO Redwoods. A rich and dynamic treebank for HPSG. *Research on Language and Computation*, *2*(4), 575–596.

Oepen, S., Kuhlmann, M., Miyao, Y., Zeman, D., Cinková, S., Flickinger, D., . . . Urešová, Z. (2015). SemEval 2015 Task 18. Broad-coverage semantic dependency parsing. In *Proceedings of the 9th International Workshop on Semantic Evaluation* (pp. 915–926). Denver, CO, USA.

Oepen, S., Kuhlmann, M., Miyao, Y., Zeman, D., Flickinger, D., Hajič, J., . . . Zhang, Y. (2014). SemEval 2014 Task 8. Broad-coverage semantic dependency parsing. In *Proceedings of the 8th International Workshop on Semantic Evaluation* (pp. 63–72). Dublin, Ireland.

Oepen, S., & Lønning, J. T. (2006). Discriminant-based MRS banking. In *Proceedings of the 5th International Conference on Language Resources and Evaluation* (pp. 1250–1255). Genoa, Italy.

Palmer, M., Gildea, D., & Kingsbury, P. (2005). The Proposition Bank. A corpus annotated with semantic roles. *Computational Linguistics*, *31*(1), 71–106.

Pan, X., Cassidy, T., Hermjakob, U., Ji, H., & Knight, K. (2015). Unsupervised entity linking with abstract meaning representation. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics.* Denver, CO, USA.

Peng, H., Thomson, S., & Smith, N. A. (2017). Deep multitask learning for semantic dependency parsing. In *Proceedings of the 55th Meeting of the Association for Computational Linguistics* (pp. 2037–2048). Vancouver, Canada.

Peng, X., Song, L., & Gildea, D. (2015). A Synchronous Hyperedge Replacement Grammar based approach for AMR parsing. In *Proceedings of the 19th Conference on Natural Language Learning* (pp. 32–41). Bejing, China.

Peng, X., Song, L., Gildea, D., & Satta, G. (2018). Sequence-to-sequence models for cache transition systems. In *Proceedings of the 56th Meeting of the Association for Computational Linguistics* (pp. 1842–1852). Melbourne, Australia.

References XV

Peng, X., Wang, C., Gildea, D., & Xue, N. (2017). Addressing the data sparsity issue in neural AMR parsing. In *Proceedings of the 15th Meeting of the European Chapter of the Association for Computational Linguistics* (pp. 366–375). Valencia, Spain.

Popel, M., Mareek, D., t ě p á nek, J., Zeman, D., & abokrtsk, Z. (2013). Coordination structures in dependency treebanks. In *Proceedings of the 51th Meeting of the Association for Computational Linguistics* (pp. 517–527). Sofia, Bulgaria.

Pourdamghani, N., Gao, Y., Hermjakob, U., & Knight, K. (2014). Aligning English strings with abstract meaning representation graphs. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing* (pp. 425–429). Doha, Qatar. doi: 10.3115/v1/D14-1048

References XVI

Prange, J., Schneider, N., & Abend, O. (2019). Semantically constrained multilayer annotation: The case of coreference. In *Proceedings of the first international workshop on designing meaning representations (dmr) at acl 2019*.

Pustejovsky, J., Lai, K., & Xue, N. (2019). Modeling quantification and scope in Abstract Meaning Representations. In *Proceedings of the first international workshop on designing meaning representations (dmr) at acl 2019*.

Reddy, S., Lapata, M., & Steedman, M. (2014). Large-scale semantic parsing without question-answer pairs. *Transactions of the Association for Computational Linguistics*.

Sagae, K., & Tsujii, J. (2008). Shift-reduce dependency DAG parsing. In *Proceedings of the 22nd International Conference on Computational Linguistics* (pp. 753–760). Manchester, UK.

Schuster, S., & Manning, C. D. (2016). Enhanced English Universal Dependencies. An improved representation for natural language understanding tasks. In *Proceedings of the 10th International Conference on Language Resources and Evaluation.* Portorož, Slovenia.

Sgall, P., Hajičová, E., & Panevová, J. (1986). *The meaning of the sentence and its semantic and pragmatic aspects*. Dordrecht, The Netherlands: D. Reidel Publishing Company.

Sun, W., Chen, Y., Wan, X., & Liu, M. (2019). Parsing Chinese sentences with grammatical relations. *Computational Linguistics*, *45*(1), 95–136. doi: 10.1162/coli_a_00343

Wang, C., Xue, N., & Pradhan, S. (2015a). Boosting transition-based AMR parsing with refined actions and auxiliary analyzers. In *Proceedings of the 53rd Meeting of the Association for Computational Linguistics and of the 7th International Joint Conference on Natural Language Processing* (pp. 857–862). Bejing, China.

Wang, C., Xue, N., & Pradhan, S. (2015b). A transition-based algorithm for AMR parsing. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics* (pp. 366–375). Denver, CO, USA.

Wang, C., Xue, N., & Pradhan, S. (2015c). A transition-based algorithm for AMR parsing. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics* (pp. 366–375). Denver, CO, USA.

White, A. S., Reisinger, D., Sakaguchi, K., Vieira, T., Zhang, S., Rudinger, R., . . . Van Durme, B. (2016). Universal Decompositional Semantics on Universal Dependencies. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing* (pp. 1713–1723). Austin, TX, USA.

Zhang, S., Ma, X., Duh, K., & Van Durme, B. (2019). Amr parsing as sequence-to-graph transduction. In *Proceedings of the 57th Meeting of the Association for Computational Linguistics.* Florence, Italy.

References XIX

Zhang, S., Ma, X., Rudinger, R., Duh, K., & Van Durme, B. (2018). Cross-lingual decompositional semantic parsing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing* (pp. 1664–1675). Brussels, Belgium.

Zhang, X., Du, Y., Sun, W., & Wan, X. (2016). Transition-based parsing for deep dependency structures. *Computational Linguistics*, *42*(3), 353–389.