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.

PETE

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

 \longrightarrow Something is high up on my dress.

PETE

PETE

 \longrightarrow Something is high up on my dress.

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

 $\xrightarrow{?}$ A group of people is on a beach.

ICK

PETE

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

 \longrightarrow Something is high up on my dress.

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

 \longrightarrow A group of people is on a beach.

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

 \longrightarrow Something is high up on my dress.

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

 \longrightarrow A group of people is on a beach.

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

 $\xrightarrow{?}$ The Commissioner was in Athens on May 2.

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

 \longrightarrow Something is high up on my dress.

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

 \longrightarrow A group of people is on a beach.

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

→ The Commissioner was in Athens on May 2.

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.

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

 \longrightarrow A group of people is on a beach.

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

 \rightarrow The Commissioner was in Athens on May 2.

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?



PETE

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

 \longrightarrow Something is high up on my dress.

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

 \longrightarrow A group of people is on a beach.

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

 $\xrightarrow{-?}$ The Commissioner was in Athens on May 2.

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.
- Logical inference or distributional approximation, both need structure.

SICK Cleo Condoravdi

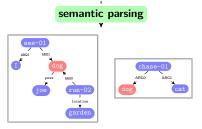


I saw Joe's dog, which was running in the garden.
The dog was chasing a cat.



I saw Joe's dog, which was running in the garden.

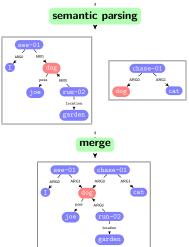
The dog was chasing a cat.





I saw Joe's dog, which was running in the garden.

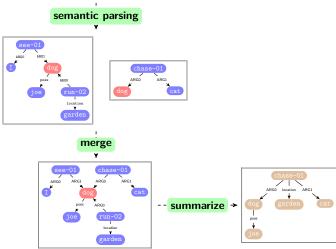
The dog was chasing a cat.





I saw Joe's dog, which was running in the garden.

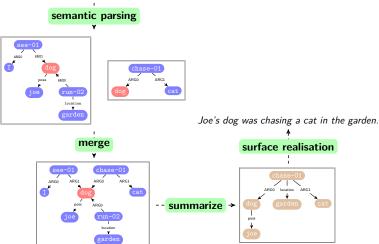
The dog was chasing a cat.





I saw Joe's dog, which was running in the garden.

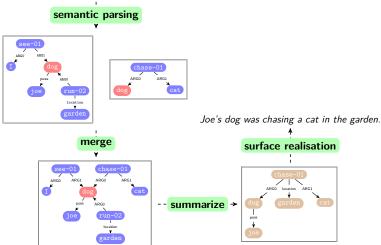
The dog was chasing a cat.





I saw Joe's dog, which was running in the garden.

The dog was chasing a cat.



Hardy & Vlachos (2018): 2⁺ ROUGE points over strong encoder–decoder.

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 output structures beyond rooted trees;
- distinct techniques, e.g. based on transitions, composition, 'translation';
- ▶ much 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;
- $\,\rightarrow\,$ clarify concepts and vocabulary; high-level survey of selected resources.

Parsing into Graph-Structured Representations

- ► Cottage industry of parsers with output structures beyond rooted trees;
- distinct techniques, e.g. based on transitions, composition, 'translation';
- ▶ much framework-internal evolution: design reflects specific assumptions;
- $\rightarrow\,$ tease apart sub-tasks and families of approaches; review representatives.

Community Fragmentation and 'Balkanization'

► Cross-Framework Perspective: Seek commonality and complementarity.

Outline: Our Game Plan



- (1) & (2) Foundations: Linguistic & Formal (0:30)
- ▶ Introduce various 'facets' (or layers) of meaning; common terminology.
- (3) Graph-Based Meaning Banks (0:45)
- Selective and terse 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)
- ▶ Introduce various 'facets' (or layers) of meaning; common terminology.
- (3) Graph-Based Meaning Banks (0:45)
- ► Selective and terse review of five English corpora with semantic graphs;
- $\,\blacktriangleright\,$ highlight distinct design decisions and goals; contrast across schools.
- (4) Parsing into Semantic Graphs (1:00)
- ► Factorization-, composition-, transition-, translation-based techniques;
- $\,\blacktriangleright\,$ 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; Please Introduce Yourself (Name & Affiliation).

Foundations: Semantics



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.

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. To answer the question precisely is difficult.

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. To answer the question precisely is difficult.

the patient's arrival the arrival by the patient the office manager the manager of the office

_

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. To answer the question precisely is difficult.

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;



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. To answer the question precisely is difficult.

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).



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)



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.



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. He_i loves his_i hat. VS. Han elsker hans hatt. He_i loves his_{$j\neq i$} hat.



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. He_i loves his_i hat. Han elsker hans hatt. He_i loves his_{$i\neq i$} hat.

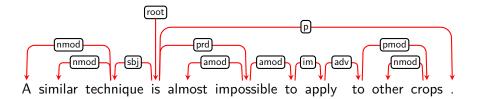


Sherlock saw the suspect with the binoculars.



Syntactic Trees vs. Semantic Graphs [WSJ #0209013]

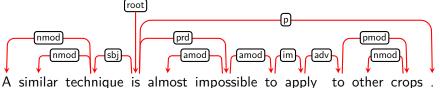




ç

Syntactic Trees vs. Semantic Graphs [WSJ#0209013]





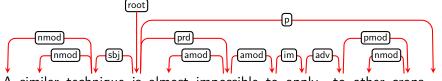
A similar technique is almost impossible to apply to other crops



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

Syntactic Trees vs. Semantic Graphs [WSJ #0209013]





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

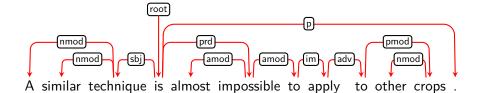


A similar technique is almost impossible to apply to other crops .

$$\begin{split} \exists x: \mathsf{technique'}(x) \land \mathsf{similar'}(x), \exists y: \mathsf{crop'}(y) \land \mathsf{other'}(y) \\ &\rightarrow \mathsf{almost'}(\neg \mathsf{possible'}(\mathsf{apply'}(_, x, y))) \end{split}$$

Syntactic Trees vs. Semantic Graphs [WSJ #0209013]





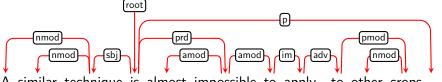
A similar technique is almost impossible to apply to other crops .

top

$$\begin{split} \exists \pmb{x} : \mathsf{technique'}(\pmb{x}) \land \mathsf{similar'}(\pmb{x}), \exists y : \mathsf{crop'}(y) \land \mathsf{other'}(y) \\ &\rightarrow \mathsf{almost'}(\neg \mathsf{possible'}(\mathsf{apply'}(_, \pmb{x}, y))) \end{split}$$

Syntactic Trees vs. Semantic Graphs [WSJ #0209013]





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

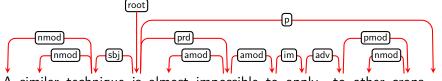


A similar technique is almost impossible to apply to other crop

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

Syntactic Trees vs. Semantic Graphs [WSJ #0209013]





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

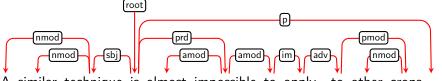


A similar technique is almost impossible to apply to other crops .

$$\begin{split} \exists x: \mathsf{technique'}(x) \land \mathsf{similar'}(x), \exists y: \mathsf{crop'}(y) \land \mathsf{other'}(y) \\ &\rightarrow \mathsf{almost'}(\neg \mathsf{possible'}(\mathsf{apply'}(_, x, y))) \end{split}$$

Syntactic Trees vs. Semantic Graphs [WSJ #0209013]





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



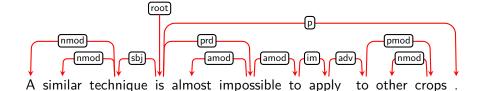
A similar technique is almost impossible to apply to other crops .

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

9

Syntactic Trees vs. Semantic Graphs [WSJ #0209013]







$$\begin{split} \exists x: \mathsf{technique'}(x) \land \mathsf{similar'}(x,_), \exists y: \mathsf{crop'}(y) \land \mathsf{other'}(y,_) \\ &\rightarrow \mathsf{almost'}(\neg \mathsf{possible'}(\mathsf{apply'}(_,x,y))) \end{split}$$

Different Desiderata (and Levels of Abstraction)

► Grammaticality (e.g. subject—verb agreement) vs. relational structure.

9



Structural Wellformedness Conditions on Trees

- ▶ Unique root, connected, single parent, free of cycles; maybe: projective;
- ightarrow all nodes (but the root) reachable by unique directed path from root.



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.





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.



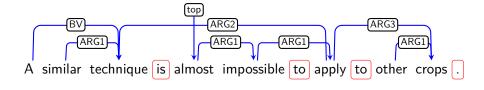
Beyond Trees: General Graphs

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



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.



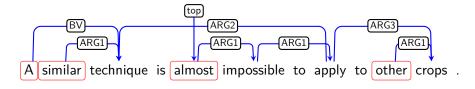
Beyond Trees: General Graphs

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



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.



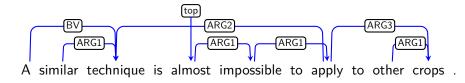
Beyond Trees: General Graphs

- ightharpoonup 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;



Structural Wellformedness Conditions on Trees

- ▶ Unique root, connected, single parent, free of cycles; maybe: projective;
- ightarrow 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;
- \rightarrow massive growth in modeling and algorithmic complexity (NP-complete).



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:



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:



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:



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:



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 each specific predicate;



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 each specific predicate;



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 each specific predicate;



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 each specific predicate;



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 each specific predicate;

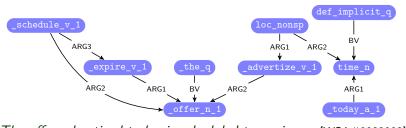
Abrams bet \$10 on Monday that it rained in Florence.

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

Reflections on Predicate-Argument Structure



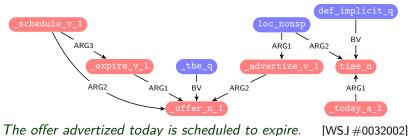
Arguments recursively are predicates most of the time (and vice versa);



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;

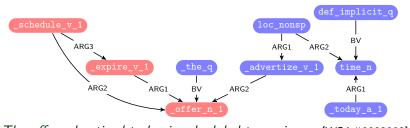


Reflections on Predicate-Argument Structure



- 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):

```
\mathsf{offer'}(x) \land \mathsf{schedule'}(\_, x, \mathsf{expire'}(x))
```



The offer advertized today is scheduled to expire.

[WSJ #0032002]

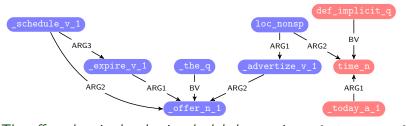
Reflections on Predicate–Argument Structure



- 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):

```
\mathsf{offer'}(x) \land \mathsf{schedule'}(\_, x, \mathsf{expire'}(x))
```

decomposition: possibly multiple predicates per word or construction.



The offer advertized today is scheduled to expire.

[WSJ #0032002]

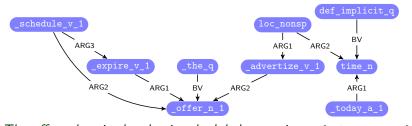
Reflections on Predicate–Argument Structure



- 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):

```
\mathsf{offer'}(x) \land \mathsf{schedule'}(\_, x, \mathsf{expire'}(x))
```

decomposition: possibly multiple predicates per word or construction.

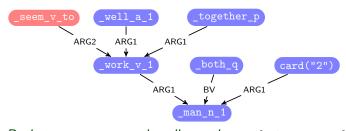


The offer advertized today is scheduled to expire.

[WSJ #0032002]

Mismatches between Syntax and Semantics

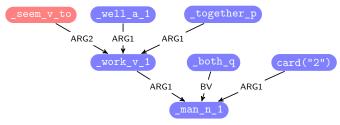




- Both men seem to work well together. [WSJ#0109043]
- Discord between head-dependent and predicate-argument relations;
- syntactic <u>subject</u> of <u>seem</u> is a semantic argument of its <u>complement</u>;

Mismatches between Syntax and Semantics

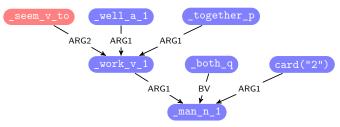




- Both men seem to work well together. [WSJ#0109043]
- ▶ Discord between head-dependent and predicate-argument relations;
- syntactic <u>subject</u> of <u>seem</u> is a semantic argument of its <u>complement</u>;
- ► consider close paraphrase: *It seems that both men work well together.*
- expletive 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 <u>subject</u> of <u>seem</u> is a semantic argument of its <u>complement</u>;
- ► 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.

Example Design Decisions: Copula Constructions

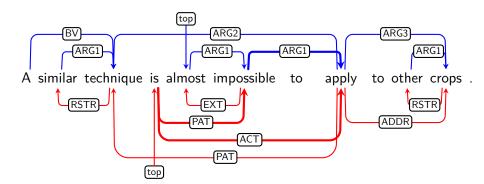


- ► There is much variation in analysis of individual linguistic phenomena;
- ► specific semantic framework requires many interacting design decisions;

Example Design Decisions: Copula Constructions

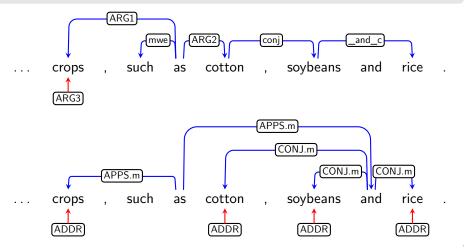


- ► There is much variation in analysis of individual linguistic phenomena;
- specific semantic framework requires many interacting design decisions;
- divergent views already at the level of which words are content-bearing;
- ▶ for example, the predicative copula: the fierce dog vs. the dog is fierce.



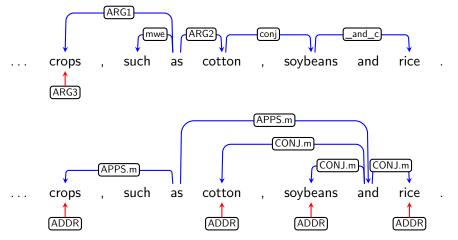
Example Design Decisions: Coordination

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



Example Design Decisions: Coordination

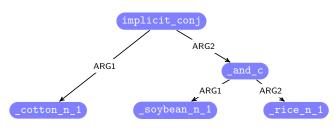
- ► No less frequent: coordinate structures (sometimes called parataxis);
- ▶ different types: Abrams arrived and ate vs. Abrams and Brown arrived.
- ▶ intuitively, a kind of 'grouping'; how to represent the group as a whole?



Further Variations on Coordination



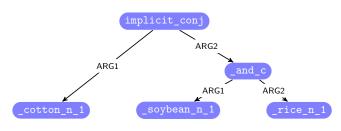
cotton, soybeans, and rice

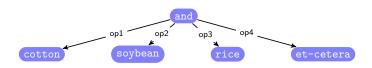


Further Variations on Coordination



cotton, soybeans, and rice

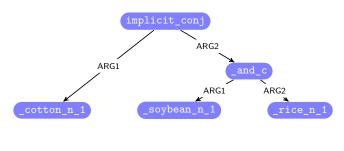


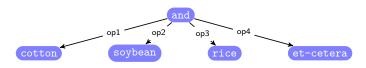


Further Variations on Coordination



cotton, soybeans, and rice





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

Background: Partial Predicate–Argument Structures



- ► 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

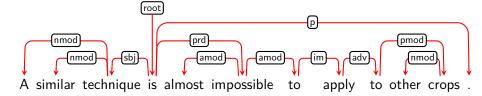


- ► Predicate—argument structure with strong focus on verbal predicates;
- ▶ large, manually annotated corpora available (for multiple languages), e.g. Framenet (Baker et al., 1998), PropBank (Palmer et al., 2005);
- ► argument entities originally syntactic constituents or simply text spans;

Background: Partial Predicate-Argument Structures



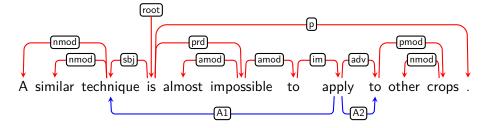
- ► Predicate—argument structure with strong focus on verbal predicates;
- ▶ large, manually annotated corpora available (for multiple languages), e.g. Framenet (Baker et al., 1998), PropBank (Palmer et al., 2005);
- argument entities originally syntactic constituents or simply text spans;
- ► 2008 Shared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



Background: Partial Predicate-Argument Structures



- ► 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:

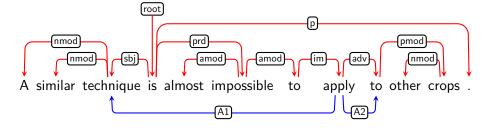


Background: Partial Predicate-Argument Structures



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:



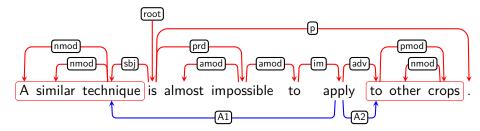
Several (semantic) predicate—argument relations remain unannotated;

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 <u>logical structure</u> 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, embedded, etc.
- 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 <u>senses</u>. 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)

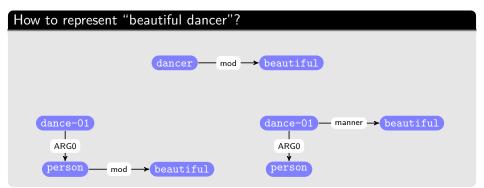
Senses of keep in Propbank

- 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.



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.

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 ...

Compositionality





(Example by Jan van Eijck)

The Principle of Compositionality

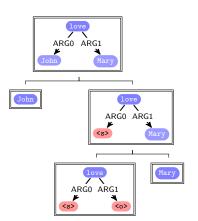
The meaning of an expression is a function of the meanings of its parts and of the way they are syntactically combined. B. Partee

Compositionality



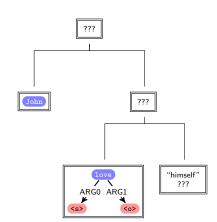
Not all semantic phenomena lend themselves easily to a compositional analysis.

Predicate—argument structure "John loves Mary."



Coreference

"John loves himself."



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;
- $ightharpoonup T \subseteq N$ 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;
- $ightharpoonup T \subseteq N$ 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;
- lacktriangledown a node n is reentrant if ${\it in}(n)>1$ (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;
- $ightharpoonup T \subseteq N$ 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;
- ightharpoonup a node n is reentrant if in(n) > 1 (shared argument across predicates);
- ightharpoonup cycles can occur: directed path from m to n and ('back') from n to m;



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

- ▶ 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;
- lacktriangledown a node n is reentrant if ${\it in}(n)>1$ (shared argument across predicates);
- ightharpoonup cycles can occur: directed path from m to n and ('back') from n to m;
- ► G is connected if there is an undirected path between all pairs of nodes;



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

- ▶ 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);
- ightharpoonup cycles can occur: directed path from m to n and ('back') from n to m;
- ▶ **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 (or Edge) Decorations



Zero-Arity Predicates vs. Constants

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

Graph Structure vs. Node (or Edge) Decorations



Zero-Arity Predicates vs. Constants

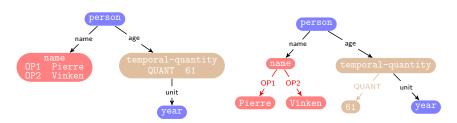
- ► Nodes and edges can be labeled (e.g. by relation and role identifiers);
- ▶ labels can be internally structured: node properties and edge attributes;
- properties (and attributes) are non-recursive attribute-value matrices;
- ▶ node (and edge) label is merely a distinguished property (or attribute);

Graph Structure vs. Node (or Edge) Decorations



Zero-Arity 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.





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



- ▶ 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;



- ▶ 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



- ▶ 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; should 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

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

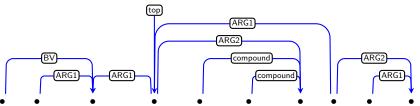
- ▶ PropBank (Palmer et al., 2005), Framenet (Baker et al., 1998), ...;
- ► Groningen Parallel Meaning Bank: GMB, PMB (Basile et al., 2012);
- ► Universal Decompositional Semantics (White et al., 2016);
- ► Enhanced Universal Dependencies (Schuster & Manning, 2016);
- Elinanced Oniversal Dependencies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



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

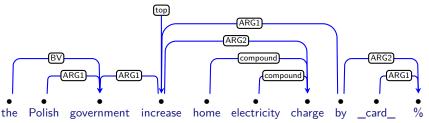


- ► Two decades of great advances in syntactic dependencies and parsing;
- ► recently, renewed interest in meaning; algorithmic interest in graphs;
- ▶ nodes limited to surface lexical units (words):



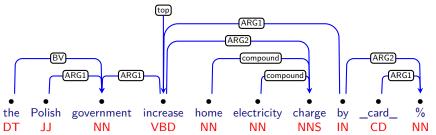


- ► Two decades of great advances in syntactic dependencies and parsing;
- recently, renewed interest in meaning; algorithmic interest in graphs;
- ▶ nodes limited to surface lexical units (words): lemmas



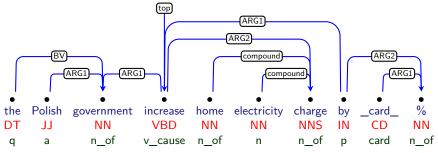


- ► Two decades of great advances in syntactic dependencies and parsing;
- ► recently, renewed interest in meaning; algorithmic interest in graphs;
- ▶ nodes limited to surface lexical units (words): lemmas, PoS



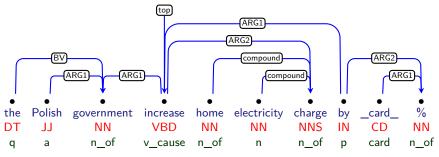


- ► Two decades of great advances in syntactic dependencies and parsing;
- recently, renewed interest in meaning; algorithmic interest in graphs;
- ▶ nodes limited to surface lexical units (words): lemmas, PoS, and frames;





- ► 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;

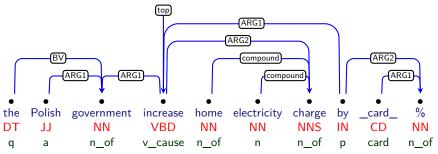


Arguably Basicest: Bilexical Semantic Dependencies



- ► 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;
- ightharpoonup 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) $^{\circ}$



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;
- ightharpoonup 34,004 + 1348 sentences, 745,543 + 29,808 tokens from WSJ corpus;
- ightharpoonup 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);
- $+ \ \ \text{frame (or sense) prediction; evaluation beyond 'atomic' dependency } \mathsf{F}_1.$

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 F₁.

+ Chinese PAS (32,783; 687,433) and Czech PSD (48,972; 1,111,626);

SemEval 2015 (Oepen et al., 2015)

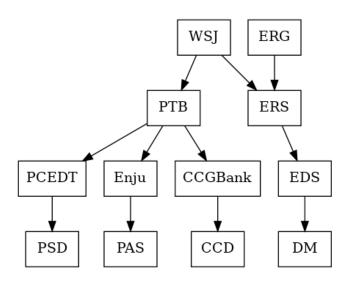
- + Out-of-domain English test data from Brown Corpus: 1,849; 31,583;
- + frame (or sense) prediction; evaluation beyond 'atomic' dependency F₁.

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

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

Bilexical Semantic Dependencies: English Genealogy

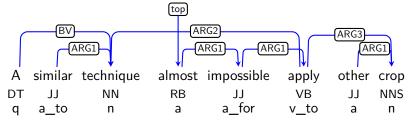




DELPH-IN MRS Bi-Lexical Dependencies (DM)



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



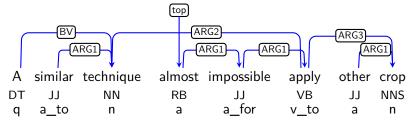
Ivanova et al. (2012)

- Simplification from underspecified logical forms (ERS; coming right up);
- ightharpoonup 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.



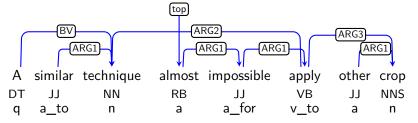
Ivanova et al. (2012)

- Simplification from underspecified logical forms (ERS; coming right up);
- ightharpoonup 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);
- ightharpoonup 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.



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;



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

- ► Hand-designed computational grammar for English in HPSG framework;
- lacktriangle 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 English Resource Grammar (Flickinger, 2000; Flickinger et al., 2017)

- ► Hand-designed computational grammar for English in HPSG framework;
- $\,\blacktriangleright\,$ 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).

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



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).

- ► 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;



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).

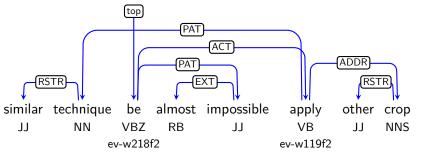
- ► Grammar-based annotation: select 'correct' reading from parse forest;
- efficient selection based on minimal discriminants; re-use across releases;
- ▶ version 1214: some 85,000 annotated sentences, six⁺ different domains;
- ▶ including Sections 00–21 from the venerable WSJ Corpus; sub-set of Brown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



- LinGO English Resource Grammar (Flickinger, 2000; Flickinger et al., 2017)
- ► Hand-designed computational grammar for English in HPSG framework;
- $\textcolor{red}{\blacktriangleright} \ \ \text{declarative, unification-based: parsing and realization; multiple engines;}$
- ▶ 25⁺ person years; coverage of 85–95 % of running text across domains;
 ▶ underspecified meaning representation in MRS (Copestake et al., 2005).

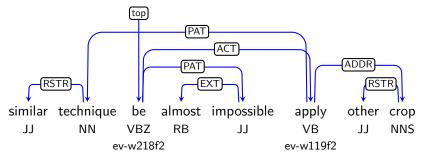
- ► Grammar-based annotation: select 'correct' reading from parse forest;
- efficient selection based on minimal discriminants; re-use across releases;
- vorcion 1214: some 85 000 apportated contanges, six+ different demains:
- version 1214: some 85,000 annotated sentences, six⁺ different domains;
 including 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.

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



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

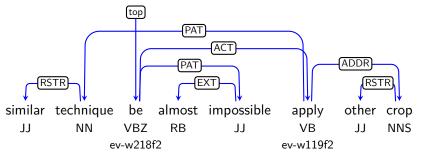
A similar technique is almost impossible to apply to other crops.



- 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;

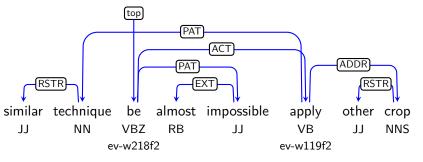


A similar technique is almost impossible to apply to other crops.



- 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);

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)

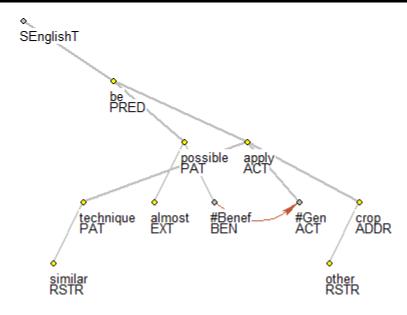
- ► Charles University in Prague, since 1960s; (Petr Sgall; 1926–2019);
- ▶ layered dependencies, e.g. morphological, analytical, tectogrammatical;
- ▶ nodes are interlinked across layers; simple feature structures on nodes;
- tectogrammatical layer: predicate—argument structure, verb senses, coreference, 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.

Background: Underlying Tectogrammatical Tree





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

Background: Tectogrammatical Valency Frames



apply² 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.

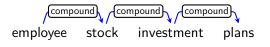
pedt A PAT similar technique PAT is almost impossible 0 * to apply *T*-1 to ADDR other crops ADDR, such as cotton ADDR, soybeans ADDR and rice ADDR.

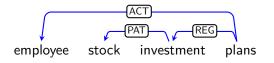
pedt Armstrong expects *-1 to close the sale of the color unit in late
November and the carpet sale in December, with the PAT gains PAT to be applied *-4 to ADDR fourth quarter or first-quarter results ADDR.

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

DM vs. PSD: Sentence vs. Speaker Meaning?







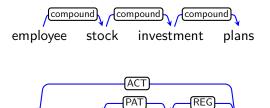
Diverging Ambitions

- ▶ 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?

emplovee





investment

plans

Diverging Ambitions

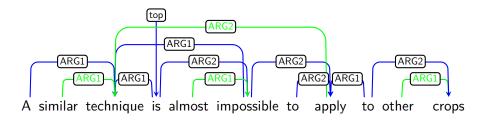
▶ Meaning determined by linguistic signal alone vs. by utterance context;

stock

- ▶ 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).

Other Bilexical 'Semantic' Dependencies (1 of 2)



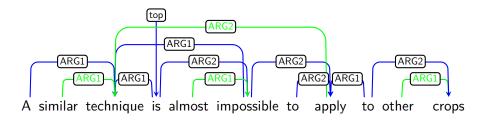


PAS: Enju Predicate–Argument Structures (Miyao, 2006)

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

Other Bilexical 'Semantic' Dependencies (1 of 2)



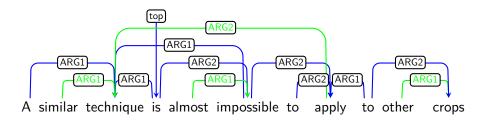


PAS: Enju Predicate–Argument Structures (Miyao, 2006)

- 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';

Other Bilexical 'Semantic' Dependencies (1 of 2)

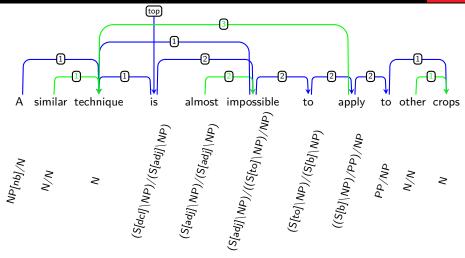




PAS: Enju Predicate–Argument Structures (Miyao, 2006)

- 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 F₁)



	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

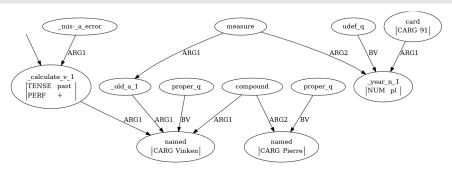
► Challenges: lexical decomposition, sub-lexical or construction semantics.

Moving On: Elementary Dependency Structures (EDS)



Limitations in Bi-Lexical Semantic Dependencies

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

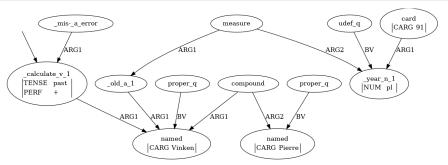


Pierre Vinken, 91 years old, had mis-calculated.

Moving On: Elementary Dependency Structures (EDS)

Limitations in Bi-Lexical Semantic Dependencies

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

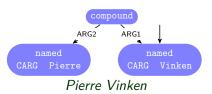


Pierre Vinken, 91 years old, had mis-calculated.

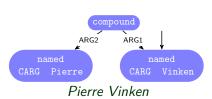
- ► ERS without variables (Oepen et al., 2002; Oepen & Lønning, 2006);
- ightharpoonup like DM, nodes encode n-ary predications, edges (expressed) arguments;
- ► Flavor (1): arbitrary (overlapping) sub-strings carry pieces of meaning.

EDSs are 'Radically Compositional'







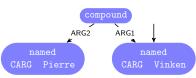


Named Entities

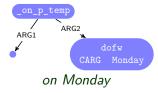
- Underspecified structure in names;
- ► few, lexically determined sub-types.

Michelle and Barack Obama





Pierre Vinken

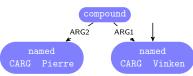


Named Entities

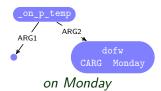
- Underspecified structure in names;
- ► few, lexically determined sub-types.

Michelle and Barack Obama





Pierre Vinken



Named Entities

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

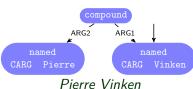
Michelle and Barack Obama

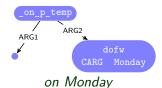
Prepositions (and Similar)

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

before and during the meeting









Named Entities

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

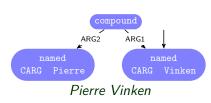
Michelle and Barack Obama

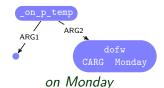
Prepositions (and Similar)

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

before and during the meeting









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







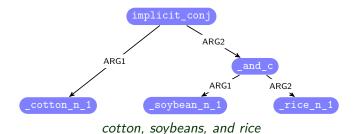


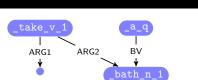
She cheerfully took a steaming-hot bath.





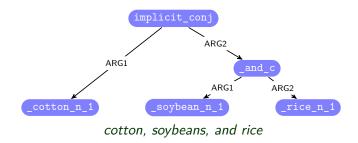
She cheerfully took a steaming-hot bath.





take a bath

She cheerfully took a steaming-hot bath.



She arrived, ate, and slept.



Sense Differentiation

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



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);



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).



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;
- $\,\blacktriangleright\,$ partial information about scope discarded in conversion to EDS graphs.



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

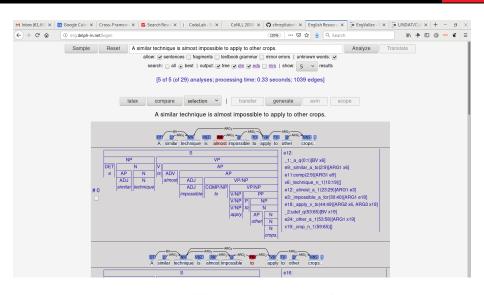
- ► Grammar-based annotation: no 'correct' analysis in 5–15 % of inputs;
- ▶ limited resources for other languages (German, Japanese, Spanish, . . .);
- ▶ tense, aspect, mood, number, etc. as 'morphological' node properties;
- ▶ partial information about scope discarded in conversion to EDS graphs.

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: Typical Smatch scores of 70–80.

Predicate-Argument Structure in AMR

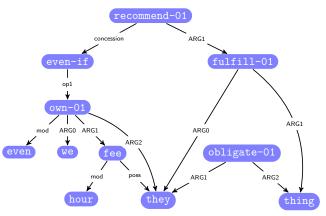




The boy wants to sleep.

More Complex AMRs



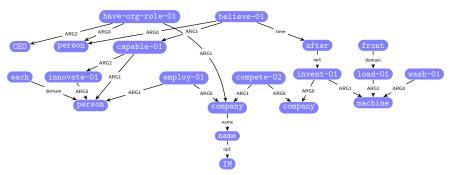


Even if we owed their hourly fees, they still should fulfill their obligations. (AMR2015 #34, simplified)

- (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 (c) coreference
- (b) named entities (d) quantification (?)

'Inverse Edges' in AMRs



Standard string format for AMRs is Penman-style:

```
want-01

ARGO ARGI

boy - ARGO — sleep-01
```

```
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:

```
snore-01 sleep-01
```

```
The man who sleeps snores.

(sn / snore-01

:ARGO (m / man

:ARGO-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.

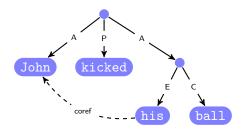
► Basic ideas:

- 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

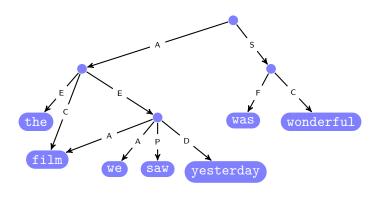
- ► A <u>scene</u> is a description of a single action or state. Sentences can contain multiple scenes. UCCA annotations distinguish between "processes" and "states".
- ▶ Scenes can have <u>participants</u> (\approx arguments) and <u>adverbials/times</u> (\approx modifiers).
- ▶ Below the clause level,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

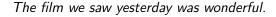
UCCA: Basic Example



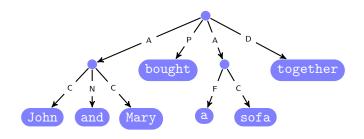
John kicked his ball.

UCCA: Argument sharing





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 (Kuhlmann & Oepen, 2016)

			DM	PSD	EDS	UCCA	AMR^{-1}
S	(01)	number of graphs	35,656	35,656	35,656	6,572	56,240
counts	(01)	number of tokens	802,717	802,717	802,717	138,268	1,000,217
ō	(02)	average number of tokens	22.51	22.51	22.51	21.03	17,78
٥	(03)	average nodes per token	0.77	0.64	1.37	1.29	0.65
	(04)	number of edge labels	59	90	15	10	101
	(05)	$%_g$ trees	2.31	42.26	34.83	0.09	22.24
	(06)	$%_g$ treewidth one	69.82	43.08	41.57	68.99	50.00
Ş	(07)	average treewidth	1.30	1.61	1.61	1.31	1.56
	(80)	maximal treewidth	3	7	4	3	5
nes	(09)	average edge density	1.019	1.073	1.053	1.015	1.092
treeness	(10)	$\%_n$ reentrant	27.43	11.41	4.98	32.78	19.89
₽	(11)	$%_g$ cyclic	0.00	0.00	0.00	0.12	0.38
	(12)	$%_g$ not connected	6.57	0.70	0.00	1.74	0.00
	(13)	$%_{g}$ multi-rooted	97.47	40.60	0.00	99.93	71.37
	(14)	percentage non-top roots	44.94	4.34	0.00	54.85	20.09
<u>_</u>	(15)	average edge length	2.684	3.320	_	_	_
order	(16)	$\%_g$ noncrossing	69.21	64.61	_	_	_
	(17)	$%_{g}$ pagenumber two	99.59	98.08	_	_	_

Parallel Meaning Bank



```
x1
person.n.01(x1)

x2 e1 t1
time.n.08(t1)
t1 = now
peel.v.01(e1)
Time(e1, t1)
Source(e1, x2)
Agent(e1, x1)
banana.n.02(x2)
```

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	土	_	±	+	_	_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

Multilingual Resources

_

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)

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

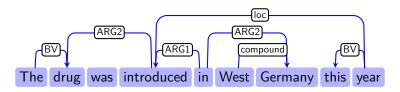
Parsing to Flavor (0) Graphs



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

	D	М	P	AS	PS	SD
	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) +Multitask learning Dozat & Manning (2018)	89.4 90.4 93.7	84.5 85.3 88.9	92.2 92.7 94.0	88.3 89.0 90.8	77.6 78.5 81.0	75.3 76.4 79.4
Lindemann et al. (2019) +Multitask learning	93.9 94.1	90.3 90.5	94.5 94.7	92.5 92.8	82.0 82.1	81.5 81.6

Accuracy of edge prediction

 $(\mathsf{id} = \mathsf{in} \; \mathsf{domain} \; \mathsf{test} \; \mathsf{set}; \; \mathsf{ood} = \mathsf{out} \; \mathsf{of} \; \mathsf{domain} \; \mathsf{test} \; \mathsf{set})$

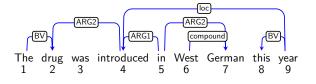
Semantic Parsing is Making Rapid Progress



_	D	М	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) +Multitask learning Dozat & Manning (2018)	89.4 90.4 93.7	84.5 85.3 88.9	92.2 92.7 94.0	88.3 89.0 90.8	77.6 78.5 81.0	75.3 76.4 79.4
Lindemann et al. (2019) +Multitask learning	93.9 94.1	90.3 90.5	94.5 94.7	92.5 92.8	82.0 82.1	81.5 81.6

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

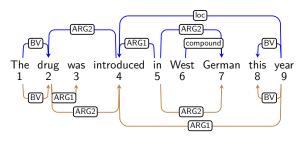




$$E_{\mathsf{gold}} = \{(1,2,\mathsf{bv}), (2,4,\mathsf{arg1}), \cdots\}$$

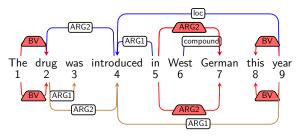
$$|E_{\mathsf{gold}}| = 7$$





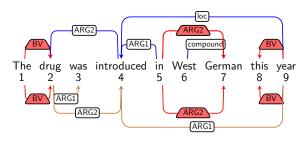
$$\begin{split} E_{\mathsf{gold}} &= \{(1,2,\mathsf{bv}), (2,4,\mathsf{arg1}), \cdots\} \\ E_{\mathsf{system}} &= \{(1,2,\mathsf{bv}), (2,3,\mathsf{arg1}), \cdots\} \end{split} \qquad \begin{aligned} |E_{\mathsf{gold}}| &= 7 \\ |E_{\mathsf{system}}| &= 6 \end{aligned}$$





```
\begin{split} E_{\mathsf{gold}} &= \{(1,2,\mathsf{bv}), (2,4,\mathsf{arg1}), \cdots\} & |E_{\mathsf{gold}}| = 7 \\ E_{\mathsf{system}} &= \{(1,2,\mathsf{bv}), (2,3,\mathsf{arg1}), \cdots\} & |E_{\mathsf{system}}| = 6 \\ E_{\mathsf{match}} &= E_{\mathsf{gold}} \cap E_{\mathsf{system}} = \{(1,2,\mathsf{bv}), (5,7,\mathsf{arg1}), \cdots\} & |E_{\mathsf{match}}| = 3 \end{split}
```





$$\begin{split} E_{\mathsf{gold}} &= \{ (1, 2, \mathsf{bv}), (2, 4, \mathsf{arg1}), \cdots \} \\ E_{\mathsf{system}} &= \{ (1, 2, \mathsf{bv}), (2, 3, \mathsf{arg1}), \cdots \} \\ E_{\mathsf{match}} &= E_{\mathsf{gold}} \cap E_{\mathsf{system}} = \{ (1, 2, \mathsf{bv}), (5, 7, \mathsf{arg1}), \cdots \} \end{split} \qquad \begin{aligned} |E_{\mathsf{gold}}| &= 7 \\ |E_{\mathsf{system}}| &= 6 \\ |E_{\mathsf{match}}| &= 3 \end{aligned}$$

$$\begin{array}{c|c} \text{Precision} & \text{Recall} & \text{F-score} \\ \hline \frac{|E_{\mathsf{match}}|}{|E_{\mathsf{system}}|} = 0.43 & \frac{|E_{\mathsf{match}}|}{|E_{\mathsf{gold}}|} = 0.5 & \frac{2*|E_{\mathsf{match}}|}{|E_{\mathsf{gold}}|+|E_{\mathsf{system}}|} = 0.46 \\ \end{array}$$

Magic Numbers Again

D	М	PAS		PSD	
id	ood	id	ood	id	ood
89.1	81.8	91.3	87.2	75.7	73.3
89.4 90.4	84.5 85.3	92.2 92.7	88.3 89.0	77.6 78.5	75.3 76.4
93.7	88.9	94.0	90.8	81.0	79.4
93.9 94.1	90.3 90.5	94.5 94.7	92.5 92.8	82.0 82.1	81.5 81.6
	id 89.1 89.4 90.4 93.7 93.9	89.1 81.8 89.4 84.5 90.4 85.3 93.7 88.9 93.9 90.3	id ood id 89.1 81.8 91.3 89.4 84.5 92.2 90.4 85.3 92.7 93.7 88.9 94.0 93.9 90.3 94.5	id ood id ood 89.1 81.8 91.3 87.2 89.4 84.5 92.2 88.3 90.4 85.3 92.7 89.0 93.7 88.9 94.0 90.8 93.9 90.3 94.5 92.5	id ood id ood id 89.1 81.8 91.3 87.2 75.7 89.4 84.5 92.2 88.3 77.6 90.4 85.3 92.7 89.0 78.5 93.7 88.9 94.0 90.8 81.0 93.9 90.3 94.5 92.5 82.0

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

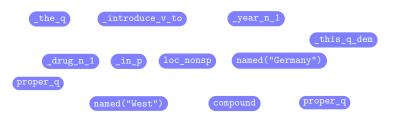
Parsing to Flavor (1) and Flavor (2) Graphs



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



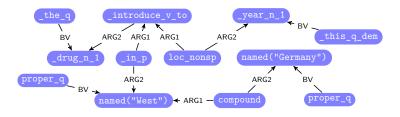
Parsing to Flavor (1) and Flavor (2) Graphs



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



	ED	S	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



	ED	S	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?



 $_{0}$ The $_{1}$ boy $_{2}$ wants $_{3}$ to $_{4}$ go $_{5}$. $_{6}$



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

 $_ ext{the}_ ext{q}\langle 0,1
angle$

 $\texttt{_want_v_to}\langle 2,3\rangle$

 $boy_n_1\langle 1,2\rangle$

 $\boxed{ _{\tt go_v_1}\langle 4,5\rangle }$



 $_{0}$ The $_{1}$ boy $_{2}$ wants $_{3}$ to $_{4}$ go $_{5}$. $_{6}$

_the_q $\langle 0,1
angle$

 $\boxed{\texttt{_want_v_to}\langle 2,3\rangle}$

 $_\mathtt{boy_n_1}\langle 1,2 \rangle$

$$\left(\operatorname{go_v_1}\langle 4,5 \rangle \right)$$

$$V_{\mathrm{gold}} = \{ (\left<0,1\right>, _\mathtt{the}_\mathtt{q}), \cdots \}, \; |V_{\mathrm{gold}}| = 4$$



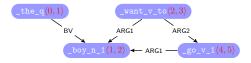
 $_{0}$ The $_{1}$ boy $_{2}$ wants $_{3}$ to $_{4}$ go $_{5}$. $_{6}$



$$V_{\mathsf{gold}} = \{(\langle 0, 1 \rangle\,, _\mathsf{the_q}), \cdots \}, \, |V_{\mathsf{gold}}| = 4$$

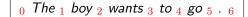


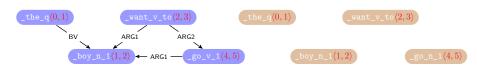
 $_{0}$ The $_{1}$ boy $_{2}$ wants $_{3}$ to $_{4}$ go $_{5}$. $_{6}$



$$\begin{split} V_{\mathrm{gold}} &= \{ (\langle 0, 1 \rangle\,,_\mathtt{the_q}), \cdots \}, \ |V_{\mathrm{gold}}| = 4 \\ E_{\mathrm{gold}} &= \{ (\langle 0, 1 \rangle\,, \mathsf{bv}, \langle 1, 2 \rangle), \cdots \}, \ |E_{\mathrm{gold}}| = 4 \end{split}$$

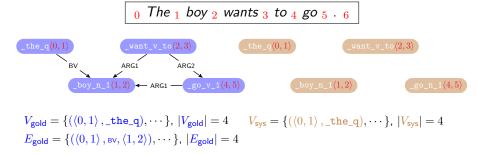




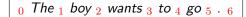


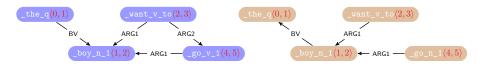
$$\begin{split} V_{\mathrm{gold}} &= \{(\langle 0,1 \rangle\,,_\mathtt{the_q}), \cdots\}, \ |V_{\mathrm{gold}}| = 4 \\ E_{\mathrm{gold}} &= \{(\langle 0,1 \rangle\,,\mathtt{BV}, \langle 1,2 \rangle), \cdots\}, \ |E_{\mathrm{gold}}| = 4 \end{split}$$





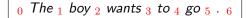


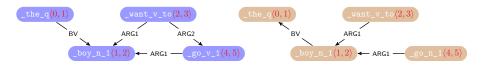




$$\begin{split} &V_{\rm gold} = \{(\left\langle 0,1\right\rangle,_{\rm the_q}),\cdots\},\ |V_{\rm gold}| = 4 \qquad V_{\rm sys} = \{(\left\langle 0,1\right\rangle,_{\rm the_q}),\cdots\},\ |V_{\rm sys}| = 4 \\ &E_{\rm gold} = \{(\left\langle 0,1\right\rangle,{}_{\rm BV},\left\langle 1,2\right\rangle),\cdots\},\ |E_{\rm gold}| = 4 \end{split}$$







$$\begin{split} &V_{\mathsf{gold}} = \{(\langle 0, 1 \rangle\,,_\mathsf{the_q}), \cdots\}, \ |V_{\mathsf{gold}}| = 4 \qquad V_{\mathsf{sys}} = \{(\langle 0, 1 \rangle\,,_\mathsf{the_q}), \cdots\}, \ |V_{\mathsf{sys}}| = 4 \\ &E_{\mathsf{gold}} = \{(\langle 0, 1 \rangle\,,\mathsf{BV}, \langle 1, 2 \rangle), \cdots\}, \ |E_{\mathsf{gold}}| = 4 \qquad E_{\mathsf{sys}} = \{(\langle 1, 2 \rangle\,,\mathsf{BV}, \langle 2, 1 \rangle), \cdots\}, \ |E_{\mathsf{sys}}| = 3 \end{split}$$



$_{0}$ The $_{1}$ boy $_{2}$ wants $_{3}$ to $_{4}$ go $_{5}$. $_{6}$

$$\begin{split} &V_{\mathsf{gold}} = \{(\langle 0, 1 \rangle\,,_\mathsf{the_q}), \cdots\}, \ |V_{\mathsf{gold}}| = 4 & V_{\mathsf{sys}} = \{(\langle 0, 1 \rangle\,,_\mathsf{the_q}), \cdots\}, \ |V_{\mathsf{sys}}| = 4 \\ &E_{\mathsf{gold}} = \{(\langle 0, 1 \rangle\,,\mathsf{bv}, \langle 1, 2 \rangle), \cdots\}, \ |E_{\mathsf{gold}}| = 4 & E_{\mathsf{sys}} = \{(\langle 1, 2 \rangle\,,\mathsf{bv}, \langle 2, 1 \rangle), \cdots\}, \ |E_{\mathsf{sys}}| = 3 \end{split}$$

$$\begin{split} V_{\mathsf{match}} &= V_{\mathsf{gold}} \cap V_{\mathsf{sys}} = \{ (\langle 1, 2 \rangle, _\mathsf{boy}_\mathsf{n}_\mathsf{1}), \cdots \} \ |V_{\mathsf{match}}| = 3 \\ E_{\mathsf{match}} &= E_{\mathsf{gold}} \cap E_{\mathsf{sys}} = \{ (\langle 2, 3 \rangle, \mathsf{ARG}, \langle 1, 2 \rangle), \cdots \} \ |E_{\mathsf{match}}| = 2 \end{split}$$



$$_{0}$$
 The $_{1}$ boy $_{2}$ wants $_{3}$ to $_{4}$ go $_{5}$. $_{6}$

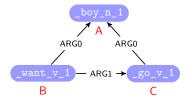
$$\begin{split} &V_{\mathsf{gold}} = \{(\left\langle 0,1\right\rangle,_\mathsf{the}_\mathsf{q}),\cdots\},\ |V_{\mathsf{gold}}| = 4 \qquad V_{\mathsf{sys}} = \{(\left\langle 0,1\right\rangle,_\mathsf{the}_\mathsf{q}),\cdots\},\ |V_{\mathsf{sys}}| = 4 \\ &E_{\mathsf{gold}} = \{(\left\langle 0,1\right\rangle,\mathsf{BV},\left\langle 1,2\right\rangle),\cdots\},\ |E_{\mathsf{gold}}| = 4 \quad E_{\mathsf{sys}} = \{(\left\langle 1,2\right\rangle,\mathsf{BV},\left\langle 2,1\right\rangle),\cdots\},\ |E_{\mathsf{sys}}| = 3 \end{split}$$

$$\begin{split} V_{\mathsf{match}} &= V_{\mathsf{gold}} \cap V_{\mathsf{sys}} = \{ (\langle 1, 2 \rangle \, , \, \mathsf{boy_n_1}), \cdots \} \, \, |V_{\mathsf{match}}| = 3 \\ E_{\mathsf{match}} &= E_{\mathsf{gold}} \cap E_{\mathsf{sys}} = \{ (\langle 2, 3 \rangle \, , \, \mathsf{aRG1}, \langle 1, 2 \rangle), \cdots \} \, \, |E_{\mathsf{match}}| = 2 \end{split}$$

$$\begin{aligned} & \mathsf{EDM_n} & \mathsf{EDM_a} & \mathsf{EDM_{na}} \\ & \frac{2*|V_{\mathsf{match}}|}{|V_{\mathsf{gold}}|+|V_{\mathsf{sys}}|} = 0.86 & \frac{2*|E_{\mathsf{match}}|}{|E_{\mathsf{gold}}|+|E_{\mathsf{sys}}|} = 0.57 & \frac{2*(|V_{\mathsf{match}}|+|E_{\mathsf{match}}|)}{|V_{\mathsf{gold}}|+|V_{\mathsf{sys}}|+|E_{\mathsf{gold}}|+|E_{\mathsf{sys}}|} = 0.67 \end{aligned}$$

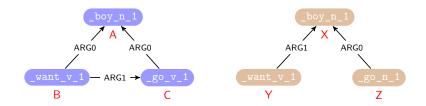


Flavor	Name	Example	Type of Anchoring
1	anchored	EDS	free node–sub-string correspondences
2	unanchored	AMR	no sub-string correspondences annotated



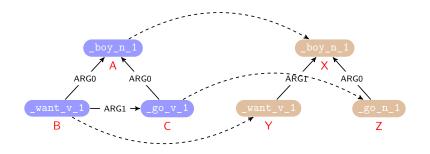


Fla	vor	Name	Example	Type of Anchoring	
	1 anchored EDS		EDS	free node-sub-string correspondences	
2	2	unanchored	AMR	no sub-string correspondences annotated	





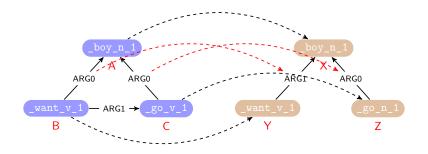
Flavor	Name	Example	Type of Anchoring	
1	1 anchored EDS		free node-sub-string correspondences	
2	unanchored	AMR	no sub-string correspondences annotated	



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



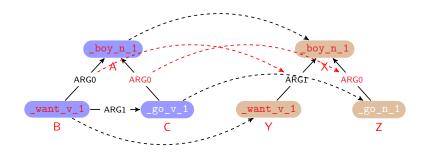
Flavor	Name	Example	Type of Anchoring	
1	1 anchored EDS		free node-sub-string correspondences	
2	unanchored	AMR	no sub-string correspondences annotated	



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



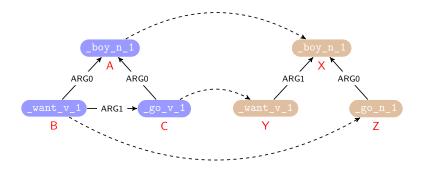
Flavor	Name	Example	Type of Anchoring	
1 anchored ED		EDS	free node–sub-string correspondences	
2 unanchored AMR		AMR	no sub-string correspondences annotated	



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



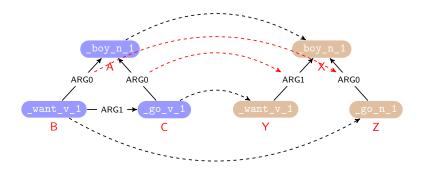
Flavor	Name	Example	Type of Anchoring	
1	1 anchored EDS		free node-sub-string correspondences	
2	unanchored	AMR	no sub-string correspondences annotated	



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



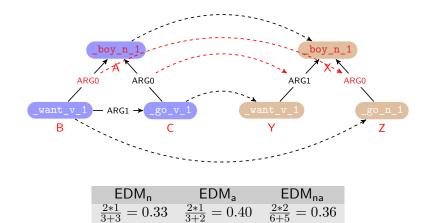
Flavor	Name	Example	Type of Anchoring	
1	1 anchored EDS		free node-sub-string correspondences	
2	unanchored	AMR	no sub-string correspondences annotated	



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

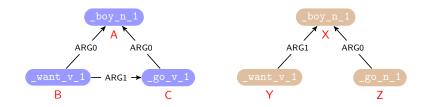


Flavor	Name	Example	Type of Anchoring	
1	1 anchored EDS		free node-sub-string correspondences	
2	unanchored	AMR	no sub-string correspondences annotated	





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



$$extstyle extstyle ext$$

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

Magic Numbers Again

+Multitask learning

	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

Modern graph parsers are cool!

90.4

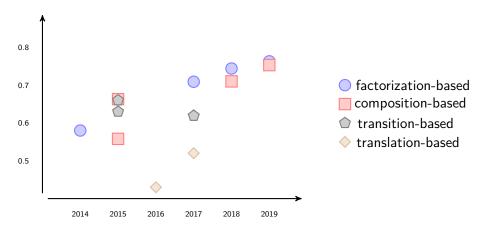
85.2

74.5

75.3

Graph Parsing is Making Rapid Progress



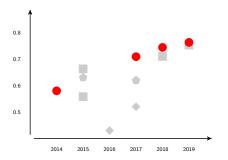


AMR parsing accuracies reported in Flanigan et al. (2014); Foland & Martin (2017); Lyu & Titov (2018); S. Zhang et al. (2019);

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



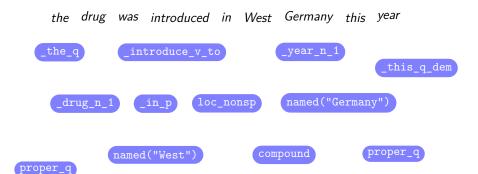


- ► Inspired by the successful design of graph-based dependency parsers. Very early work: McDonald & Pereira (2006).
- ► Explicitly modeling the target structure.
- ► A parser knows how to evaluate the *goodness* of a candidate graph.
- ► A parser knows how to find the *best* graph from an extremely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

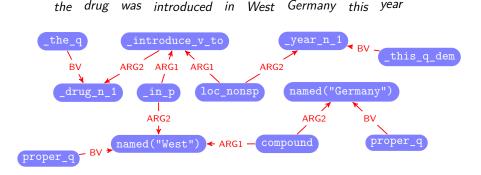


the drug was introduced in West Germany this year

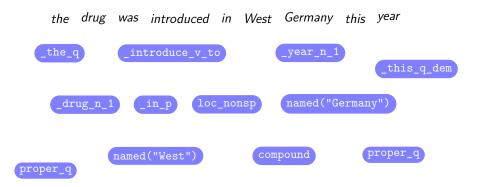












Task 1: Concept Identification



Task 1: Concept Identification



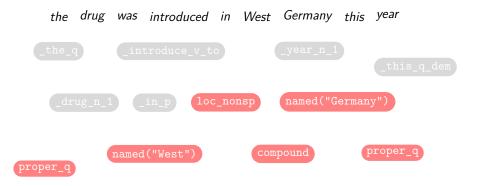


Task 1: Concept Identification

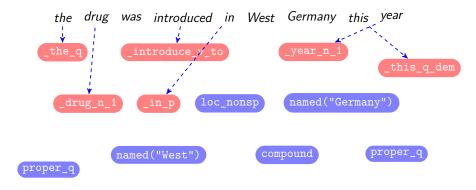


Task 1: Concept Identification

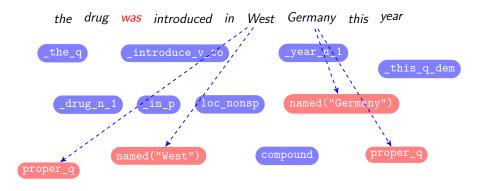




Task 1: Concept Identification

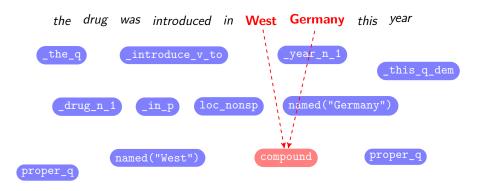


Task 0: Concept-to-word Alignment

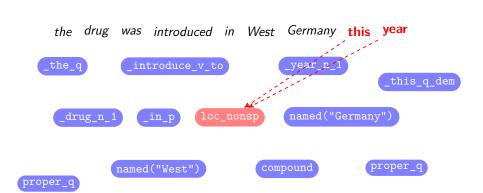


Task 0: Concept-to-word Alignment





Task 0: Concept-to-word Alignment



Task 0: Concept-to-word Alignment





- **Task 0**: Concept-to-word Alignment
- Task 1: Concept Identification
- Task 2: Relation Detection





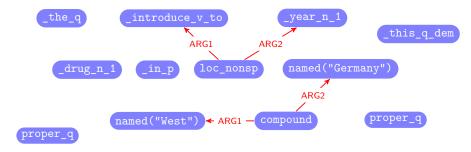
Task 0: Concept-to-word Alignment

Task 1: Concept Identification

Task 2: Relation Detection



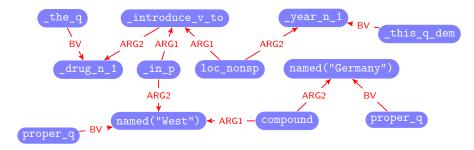
the drug was introduced in West Germany this year



- Task 0: Concept-to-word Alignment
- Task 1: Concept Identification
- Task 2: Relation Detection



the drug was introduced in West Germany this year



- Task 0: Concept-to-word Alignment
- Task 1: Concept Identification
- Task 2: Relation Detection

- ► 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 (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 $x = w_0, \dots w_{n-1}$ 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:

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

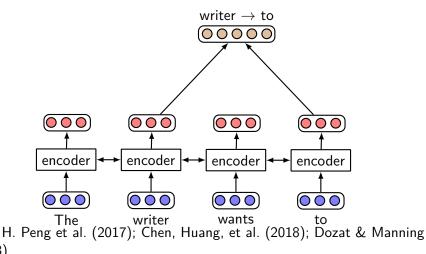
First-order factoriztion

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

 $\mbox{M.}$ Kuhlmann and P. Jonsson. 2015. Parsing to Noncrossing Dependency Graphs.

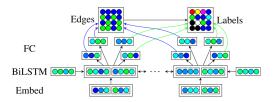


$$G' = \arg\max_{G^* = (V, E^* \subseteq E)} \sum_{e \in E^*} \text{ScorePart}(e)$$



(2018)





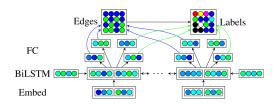
Concatenate word and POS tag embeddings.

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

Then BiLSTM them:

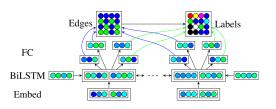
$$\mathbf{r}_1, \mathbf{r}_2, ..., \mathbf{r}_n = \text{BiLSTM}(\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_n)$$

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



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

$$\begin{array}{ccc} \mathbf{h}_i^{\mathsf{edge-head}} & = & {}_{\mathsf{FNN}}\mathsf{edge-head}(\mathbf{r}_i) \\ \\ \mathbf{h}_i^{\mathsf{edge-dep}} & = & {}_{\mathsf{FNN}}\mathsf{edge-dep}(\mathbf{r}_i) \\ \\ \mathsf{SCOREEDGE}(s,i,j) & = & \mathsf{BIAFFINE}\mathsf{edge}(\mathbf{h}_i^{\mathsf{edge-head}},\mathbf{h}_j^{\mathsf{edge-dep}}) \end{array}$$



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

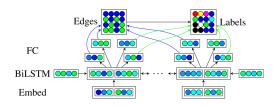
$$\begin{array}{cccc} \mathbf{h}_{i}^{\mathsf{edge-head}} & = & {}_{\mathsf{FNN}}^{\mathsf{edge-head}}(\mathbf{r}_{i}) \\ \\ \mathbf{h}_{i}^{\mathsf{edge-dep}} & = & {}_{\mathsf{FNN}}^{\mathsf{edge-dep}}(\mathbf{r}_{i}) \\ \\ \mathsf{SCOREEDGE}(s,i,j) & = & \mathsf{BIAFFINE}^{\mathsf{edge}}(\mathbf{h}_{i}^{\mathsf{edge-head}}, \mathbf{h}_{j}^{\mathsf{edge-dep}}) \end{array}$$

FNN and BIAFFINE functions are popular

BIAFFINE
$$(\mathbf{x}_1, \mathbf{x}_2) = \mathbf{x}_1^{\top} \mathbf{U} \mathbf{x}_2 + \mathbf{W}(\mathbf{x}_1 \oplus \mathbf{x}_2) + \mathbf{b}$$

FNN $(\mathbf{x}) = \text{ReLU}(\mathbf{W}\mathbf{x} + \mathbf{b})$





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

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

$$\begin{array}{lll} {\rm ScoreLabel}({i,j}) & = & {\rm Biaffine}^{\sf label}(\mathbf{h}_i^{\sf label-head}, \mathbf{h}_j^{\sf label-dep}) \\ & \mathbf{h}_i^{\sf label-head} & = & {\rm FNN}^{\sf label-head}(\mathbf{r}_i) \\ & \mathbf{h}_i^{\sf label-dep} & = & {\rm FNN}^{\sf label-dep}(\mathbf{r}_i) \\ \end{array}$$

Structure validation



Parsing to dependency trees

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

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

- $ightharpoonup E' \subseteq E$
- ightharpoonup 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 ...: $O(n^4)$

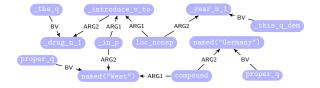
Kuhlmann & Jonsson (2015); Cao et al. (2017a, 2017b)

- ► 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)	✓	✓	✓



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





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



The	dr	ıg /	was	introduced	in		West	-	Germany	-	this	year
_the_q	_drug	_n_1	Ø	_introduce_v_to	_in_p	1	named("W") proper_q	•	named("G") proper_q		_this_q_dem	_year_n_1
							com	рот	und		loc_n	onsp



The	drug	was	introduced		in	West		Germany	this	у	ear
the_q	drug_n_1	Ø	_introduce_v_to	ľ	_in_p	named("W") proper_q		named("G") proper_q	_this_q_dem	_yea	r_n_1
						comp	01	ind	loc_no	onsp	

Almost Sequence Labeling

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

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



The	drug	was	introduced	in	West	Germany	this	year
_the_q	_drug_n_1	Ø	_introduce_v_to	_in_p	<pre>named("W") proper_q</pre>	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.

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



The	drug	was	introduced	in	West	Germany	this	year
_the_q	_drug_n_1	Ø	_introduce_v_to	_in_p	named("W") proper_q	<pre>named("G") proper_q</pre>	_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.

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



The	drug	was	introduced		in		West		Germany	this		year
_the_q	_drug_n_1	Ø	_introduce_v_to	1	_in_p		named("W") proper_q		<pre>named("G") proper_q</pre>	_this_q_dem		_year_n_1
						Ī	com	po	und	loc_no	on	sp

Almost Sequence Labeling

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

- ▶ 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	rug_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



Solution

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





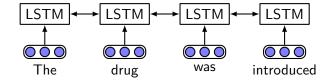






Solution

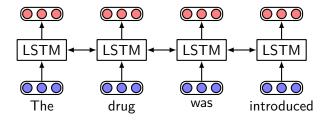
- ► Pretraining: ELMo, BERT, etc.
- ► Word encoder: LSTM, Transformer, etc.
- Classification





Solution

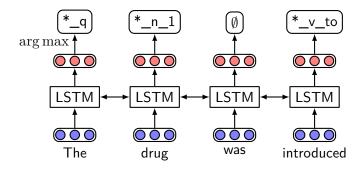
- ► Pretraining: ELMo, BERT, etc.
- ► Word encoder: LSTM, Transformer, etc.
- Classification





Solution

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



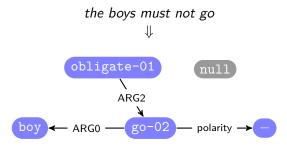
Sub-Tasks

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

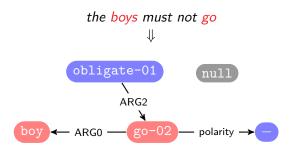
	Alignment	Concept Identification	Relation Detection
Flavor (0)			✓
Flavor (1)		✓	✓
Flavor (2)	~	✓	✓

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









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)



the boys must not go

obligate-01

go-02

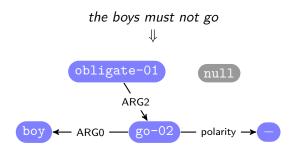
boy



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)





No annotations for concept-to-word alignment

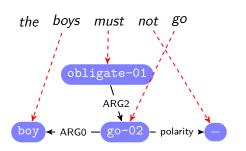
- ► Heuristic rules (Flanigan et al., 2014)
- ► Linearize graphs and reuse word alignment tools, e.g. GIZA++ and BerkeleyAligner, etc. (Pourdamghani et al., 2014)
- Consider all possible alignments (Lyu & Titov, 2018)



the boys must not go

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

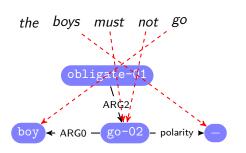
- 1. the concept identification model: $P_{\theta}(\mathbf{c}|\mathbf{a},\mathbf{w})$
- 2. the relation identification model: $P_{\phi}(R|\mathbf{a}, \mathbf{w}, \mathbf{c})$
- 3. the alignment model: $Q_{\psi}(\mathbf{a}|\mathbf{c},R,\mathbf{w})$
- C. Lyu and I. Titov. 2018. AMR Parsing as Graph Prediction with Latent Alignment



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

- 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

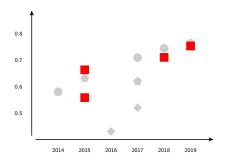


$$P_{\theta,\phi}(\mathbf{c},R|\mathbf{w}) = \sum_{\mathbf{a}} P(\mathbf{a}) P_{\theta}(\mathbf{c}|\mathbf{a},\mathbf{w}) P_{\phi}(R|\mathbf{a},\mathbf{w},\mathbf{c})$$

- 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

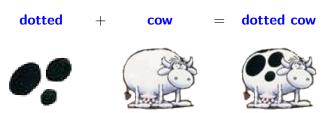




- ► Inspired by *old school*, rule-based approaches.
- Explicitly modeling the syntactico-semantic derivation process.
- ▶ A parser knows how to evaluate the *goodness* of a derivation process.
- ▶ A parser knows how to find the *best* derivation from a large set of derivations that are licensed by a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

Reminder: Compositionality





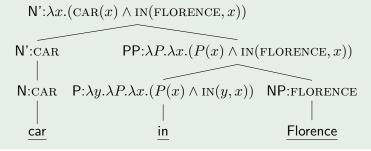
(Example from Jan van Eijck)

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

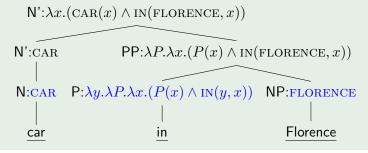


Tree construction + Lexical interpretation \Rightarrow Meaning representation



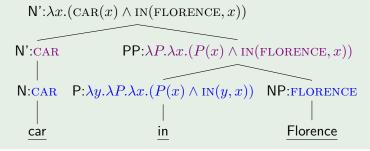


Tree construction + Lexical interpretation \Rightarrow Meaning representation



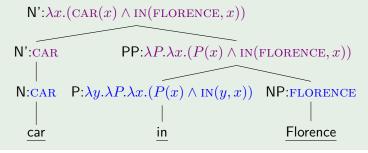


Tree construction + Lexical interpretation \Rightarrow Meaning representation





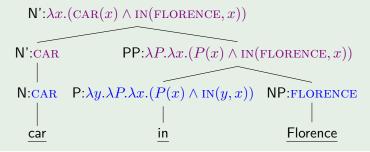
Tree construction + Lexical interpretation \Rightarrow Meaning representation





Tree construction + Lexical interpretation \Rightarrow Meaning representation

Using λ Expressions



Manipulating Graphs

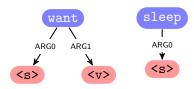
- ► Graph algebra (AM algebra)
- Graph grammar (hyperedge replacement grammar)



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")



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)



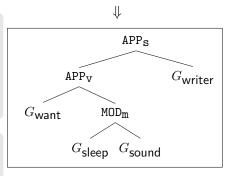
AM algebra provides a systematic way to construct graphs

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

Two operations for combining s-graphs:

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

Groschwitz et al. (2017)





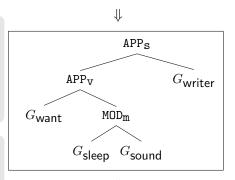
AM algebra provides a systematic way to construct graphs

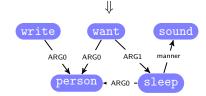
- term (=tree of operation symbols)
- ► value (=s-graph)

Two operations for combining s-graphs:

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

Groschwitz et al. (2017)







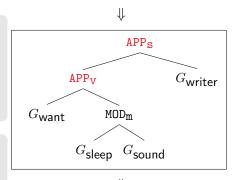
AM algebra provides a systematic way to construct graphs

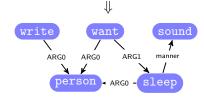
- term (=tree of operation symbols)
- ► value (=s-graph)

Two operations for combining s-graphs:

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

Groschwitz et al. (2017)







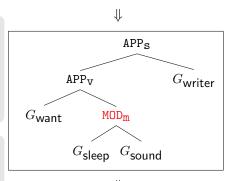
AM algebra provides a systematic way to construct graphs

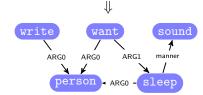
- term (=tree of operation symbols)
- ► value (=s-graph)

Two operations for combining s-graphs:

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

Groschwitz et al. (2017)

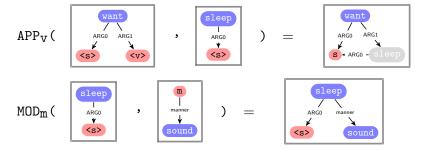




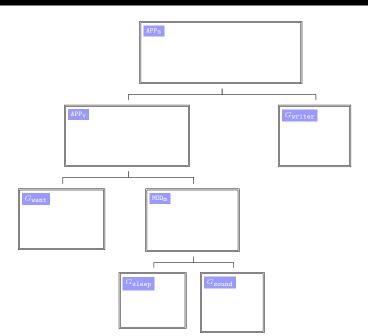


Two operations for combining s-graphs:

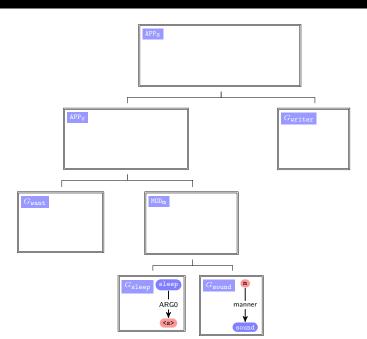
Apply (=head+complement), Modify (=head+modifier)



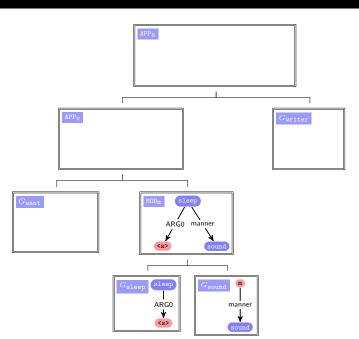
J. Groschwitz, M. Fowlie, M. Johnson and A. Koller. 2017. A constrained graph algebra for semantic parsing with AMRs.



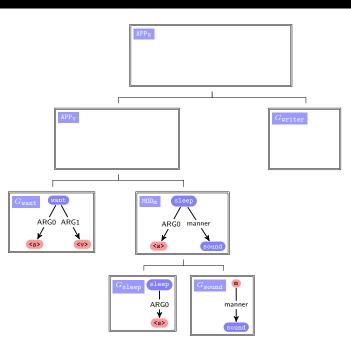




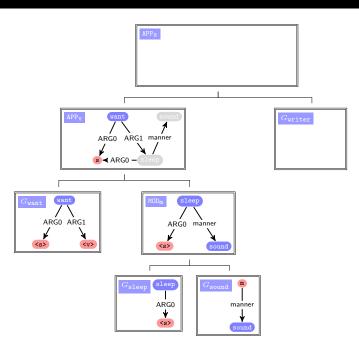




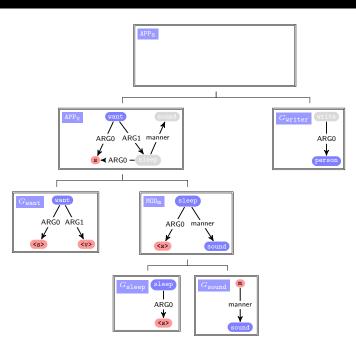




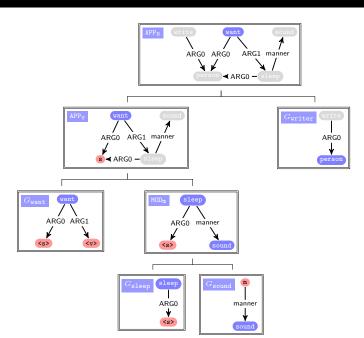










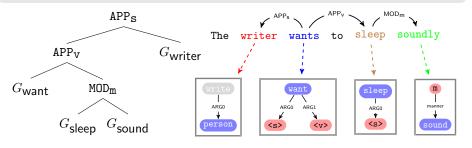




AM Algebra-Based Parsing



Tree construction + Lexical interpretation \Rightarrow 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 \Rightarrow Meaning representation

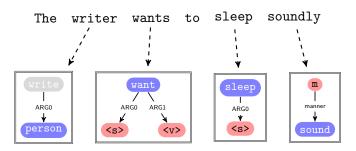
- ► A supertagger predicts graphs for words.
- ► A dependency parser predicts APP and MOD edges.
- ► At test time, compute highest-scoring well-typed AM dependency tree for input sentence.

AM Algebra-Based Parsing



Tree construction + Lexical interpretation \Rightarrow 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.

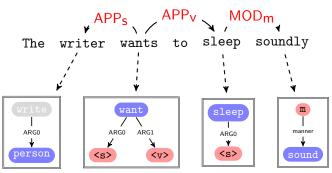


AM Algebra-Based Parsing



Tree construction + Lexical interpretation \Rightarrow 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.

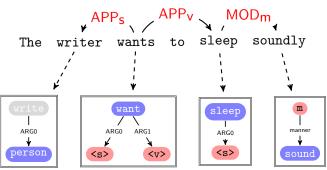


AM Algebra-Based Parsing



Tree construction + Lexical interpretation \Rightarrow 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.

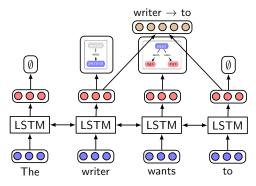


AM Algebra-Based Parsing



Tree construction + Lexical interpretation \Rightarrow 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.



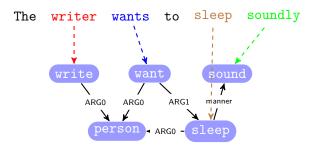


The writer wants to sleep soundly



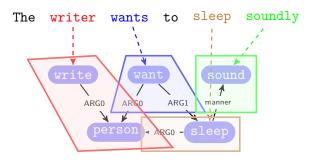
Decompose graph into parts based on concept-to-word alignment.





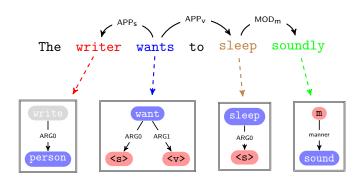
Decompose graph into parts based on concept-to-word alignment.





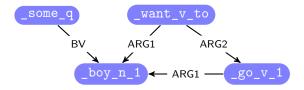
Decompose graph into parts based on concept-to-word alignment.





Training data = supertags + AM dependency tree

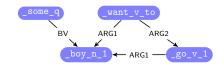
Hypergraph

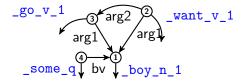


A graph consists of:

- ► 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.



↓s

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



$$\mathring{\downarrow} S \stackrel{\gamma_1}{\Rightarrow} {\text{arg1}} \bigvee_{\mathsf{NP}} {\text{arg1}}$$

- ► 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 .



$$\downarrow S \xrightarrow{\gamma_1} \underset{NP}{\operatorname{arg1}} \xrightarrow{\gamma_3} \underset{DET}{\underbrace{VP}} \underset{NNS}{\operatorname{arg1}}$$

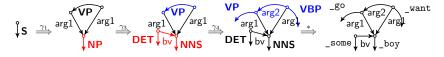
- Terminal vs. non-terminal hyperedges
- ► Non-terminal hyperedges are utilized to control a derivation process.
- ► A derivation starts from a non-terminal hyperedge .
- ► In a derivation step, we substitute a non-terminal hyperedge with a hypergraph .





- ► 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 .





- ► 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.





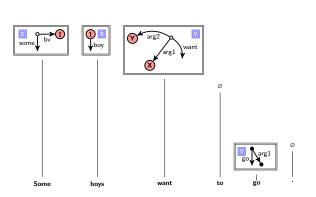
- ► Terminal vs. non-terminal hyperedges (symbols)
- ► Non-terminal hyperedges (symbols) are utilized to control a derivation process.
- ► A derivation starts from a non-terminal hyperedge (symbol).
- ► In a derivation step, we substitute a non-terminal hyperedge (symbols) with a hypergraph (a sequence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

Drewes et al. (1997)



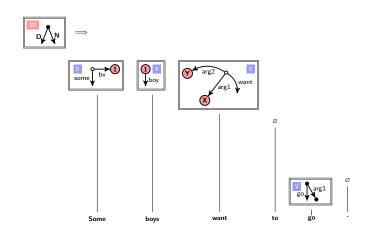
Some boys want to go

107

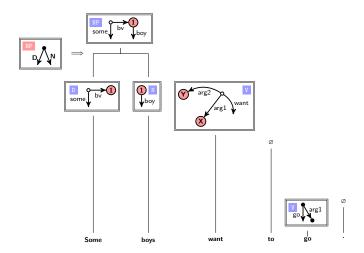




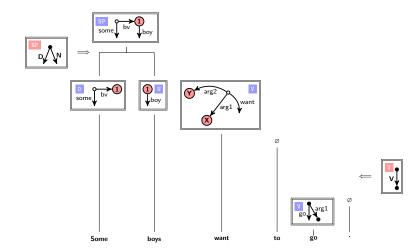




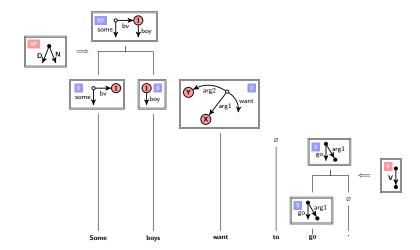




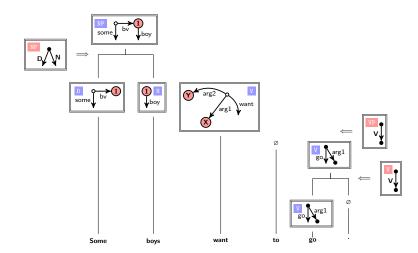




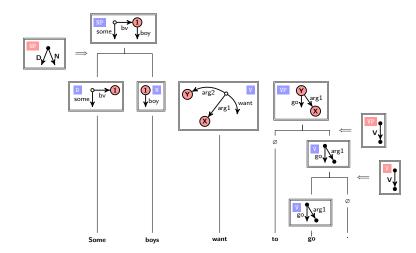




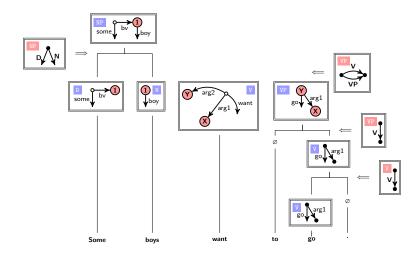




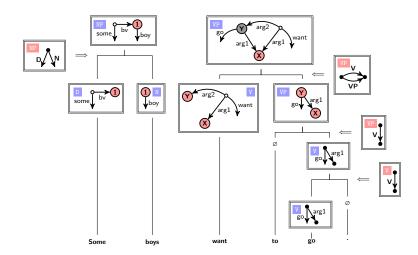




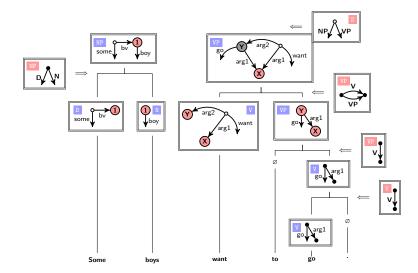




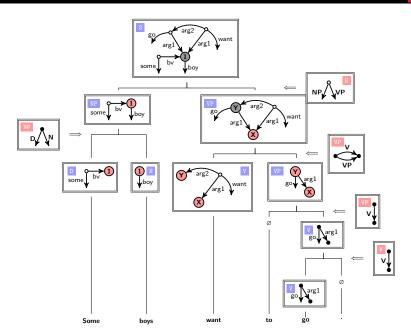










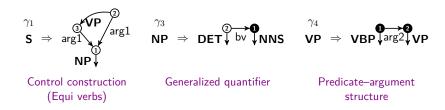


HRGs Can be Linguistically Meaningful

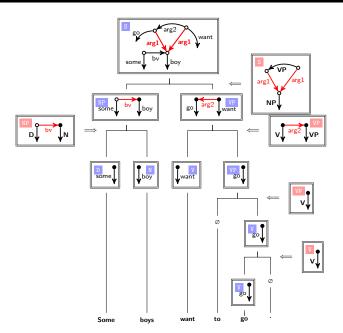




Rules



Construction Semantics (Revisited)



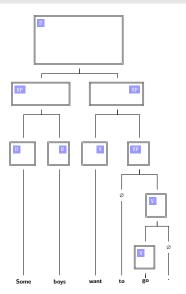




Tree construction + Semantic interpretation \Rightarrow Meaning representation

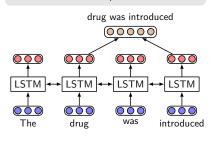


Tree construction + Semantic interpretation ⇒ Meaning representation



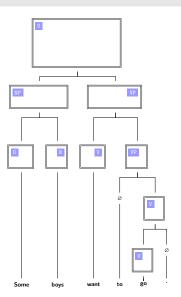
Syntactic parsing

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



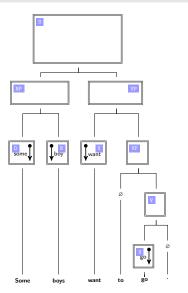


Tree construction + Semantic interpretation \Rightarrow Meaning representation





Tree construction + Semantic interpretation ⇒ Meaning representation



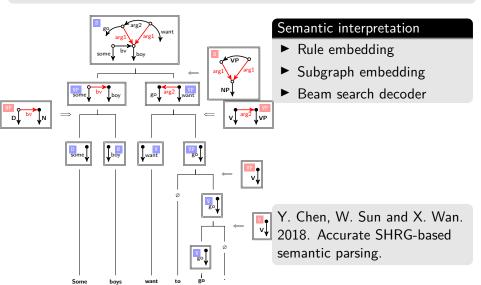
Semantic interpretation

- ► Rule embedding
- ► Subgraph embedding
- ► Beam search decoder

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



Tree construction + Semantic interpretation \Rightarrow Meaning representation



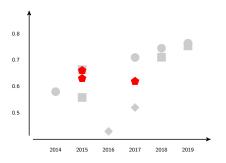
Magic Numbers



	DM		PAS		PSD		EDS		AMR	
	id	ood	id	ood	id	ood	Smatch	EDM_{na}	${\sf Smatch}$	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.



- 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

PARSE
$$(x = (w_0, w_1, \dots, w_n))$$

- 1 $c \leftarrow c_s(x)$
- 2 while $c \notin C_t$
- 3 c = Act(c, GetTransition(c))
- 4 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)



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



Partial analysis

_some_q

Transition

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

Stack

Buffer

 $[\underline{some}_q, boys, want, to, go]$

Transition Init(_some_q)

>replace Some in the buffer with _some_q



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 <u>some</u>q from the buffer to the stack;



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)

bys in the buffer with _boy_n_1



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



Partial analysis



Transition

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

Stack [_some_q]

Buffer

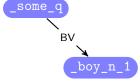
[_boy_n_1, want, to, go]

```
Transition Right-Arc(BV)
```

⊳link _some_q and _boy_n_1



Partial analysis



Transition

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

Stack [_some_q]

Buffer

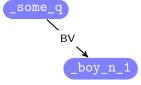
[_boy_n_1, want, to, go]

Transition Reduce

⊳remove <u>some</u> q from the stack



Partial analysis



Transition

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

Stack [] Buffer

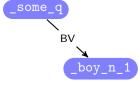
[_boy_n_1, want, to, go]

Transition Reduce

⊳remove <u>some</u>q from the stack



Partial analysis



Transition

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

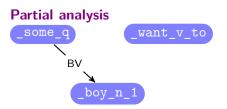
Stack [] Buffer

[_boy_n_1, want, to, go]

Transition Shift(_want_v_to)

⊳move _boy_n_1 from the buffer to stack;





Transition

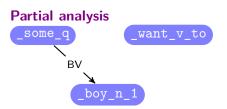
- ► Shift (CI)
- ► Left-Arc (RD)
- ► Right-Arc (RD)
- ► Reduce

```
Stack Buffer [_want_v_to, to, go]
```

```
Transition
Shift(_want_v_to) 

>then replace want in the buffer with _want_v_to
```





Transition

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

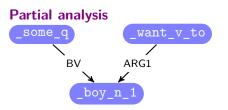
```
Stack [_boy_n_1]
```

Buffer [_want_v_to, to, go]

```
Transition Left-Arc(ARG1)
```

 \triangleright link _boy_n_1 and _want_v_to





Transition

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

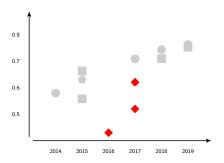
```
Stack Buffer [_want_v_to, to, go]
```

Transition Left-Arc(ARG1)

⊳link _boy_n_1 and _want_v_to

Translation-Based Approaches

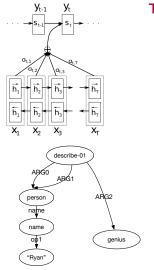




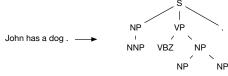
- ► Inspired by neural machine translation.
- Semantic graph as a foreign language.
- ► A parser knows how to linearize a graph.
- ▶ Data augmentation has been shown very helpful, partially reflecting the data-hungry nature of seq2seq models.

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 ARGO(person name(name op1("Ryan")op1)name)ARGO 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



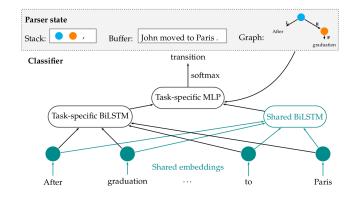
- ► State-of-the-art graph parsers rely on large-scale, manually annotated graphbanks.
- ► Multiple heterogeneous graphbanks EXIST!
- ?> Can we get a better parser by integrating heterogeneous graphbanks?
- 1. Heterogeneous annotations are (similar but) different.
 - ► Different projects, different linguistic theories, different representation formalisms, different annotation schemes, etc.
 - → 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-Do	main	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) +MTL		84.3 85.7	91.4 92.2	86.6 88.0		74.5 76.2	82.5 83.3	69.2 70.4
Single (BERT) +MTL			94.5 94.7		82.0 82.1		84.9 85.2	74.3 74.5

Magic Numbers



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

	In-Do	omain	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) +MTL			91.4 92.2			74.5 76.2	82.5 83.3	69.2 70.4
Single (BERT) +MTL		90.3 90.5	94.5 94.7		82.0 82.1		84.9 85.2	74.3 74.5

Cross-lingual parsing

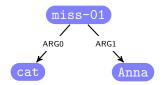


Multilingual Parsing

- ► 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

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.

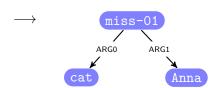
Example

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.

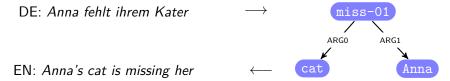
Example

DE: Anna fehlt ihrem Kater



- 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





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



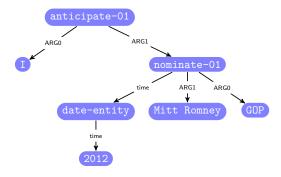
Task: Map entities in text to entities in a knowledge base (KB).

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



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.





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)

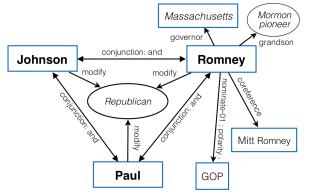


Task: Map entities in text to entities in a knowledge base (KB).

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 ...



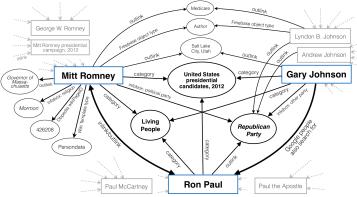


Task: Map entities in text to entities in a knowledge base (KB).

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 ...



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)

Abstractive Summarization



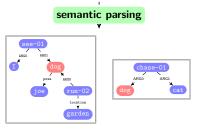
I saw Joe's dog, which was running in the garden.
The dog was chasing a cat.

Abstractive Summarization



I saw Joe's dog, which was running in the garden.

The dog was chasing a cat.

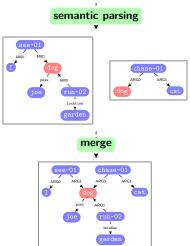


Abstractive Summarization



I saw Joe's dog, which was running in the garden.

The dog was chasing a cat.



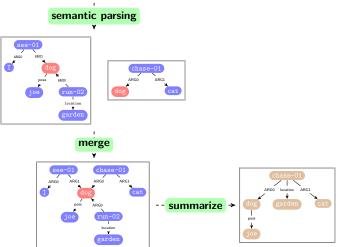
Liu et al. (2015); Hardy & Vlachos (2018)

Abstractive Summarization



I saw Joe's dog, which was running in the garden.

The dog was chasing a cat.



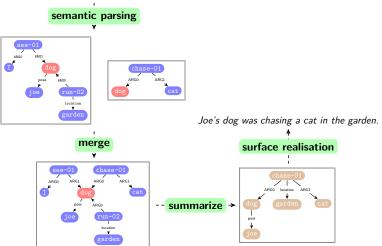
Liu et al. (2015); Hardy & Vlachos (2018)

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: Evaluation



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)

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.
- ► To what degree do they generalize across frameworks?
- ► 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



We gratefully acknowledge contributions, comments, and assistance by numerous colleagues, including:

Emily M. Bender, Dan Flickinger, and the DELPH-IN community at large;
Omri Abend and Daniel Hershcovich;
Noortje Venhuizen;
Tim O'Gorman, Martha Palmer, and Nianwen Xue;
The Chinese Student; and others.

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

Acknowledgments



We gratefully acknowledge contributions, comments, and assistance by numerous colleagues, including:

Emily M. Bender, Dan Flickinger, and the DELPH-IN community at large;
Omri Abend and Daniel Hershcovich;
Noortje Venhuizen;
Tim O'Gorman, Martha Palmer, and Nianwen Xue;
The Chinese Student; and others.

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

There might still be time for 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. Abeill é (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.

References VI

- 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.

References VIII

- 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)

References IX

- 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.

References XI

- 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.

References XIII

- 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.

References XIV

- 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., $t \\ end{e} p \\ and nek, J., Zeman, D., & abokrtsk, Z. (2013).$ 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.

References XVII

- 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.

References XVIII

- 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.