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.

→ 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

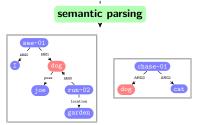


I saw Joe's dog, which was running in the garden.
The dog was chasing a cat.



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

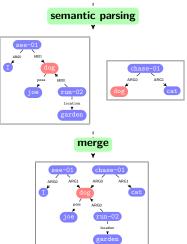
The dog was chasing a cat.





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

The dog was chasing a cat.

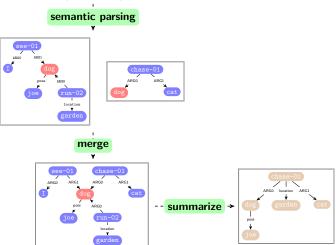


(Liu et al., 2015; Hardy & Vlachos, 2018)



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

The dog was chasing a cat.

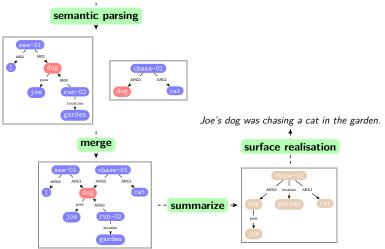


(Liu et al., 2015; Hardy & Vlachos, 2018)



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

The dog was chasing a cat.



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

High-Level Goals of the Tutorial



Graph-Based Representations of Meaning

- ► Vast, complex landscape of representing natural language meaning;
- diverse linguistic traditions, modeling assumptions, levels of ambition;
- ▶ some differences are superficial (e.g. terminology), others run deeper;
- $\,\rightarrow\,$ clarify concepts and vocabulary; high-level survey of selected resources.

High-Level Goals of the Tutorial



Graph-Based Representations of Meaning

- ► Vast, complex landscape of representing natural language meaning;
- ▶ diverse linguistic traditions, modeling assumptions, levels of ambition;
- ▶ some differences are superficial (e.g. terminology), others run deeper;
- $\,\rightarrow\,$ clarify concepts and vocabulary; high-level survey of selected resources.

Parsing into Graph-Structured Representations

- Cottage industry of parsers with outputs structures beyond rooted trees;
- distinct techniques, e.g. based on transitions, composition, 'translation';
- ▶ some framework-internal evolution: design reflects specific assumptions;
- ightarrow 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 outputs structures beyond rooted trees;
- distinct techniques, e.g. based on transitions, composition, 'translation';
- $\textcolor{red}{\blacktriangleright} \ \ \text{some framework-internal evolution: design reflects specific assumptions;}$
- $\rightarrow\,$ tease apart sub-tasks and families of approaches; review representatives.

Fragmentation and 'Balkanization'

► Cross-Framework Perspective: Seek commonality and complementarity.

Outline: Our Game Plan



Foundations: Linguistic & Formal (0:30)

► Tease apart various 'facets' (layers) of meaning; common terminology.

Graph-Based Meaning Banks (0:45)

- ► Semi-superficial review of five English corpora with semantic graphs;
- ▶ highlight distinct design decisions and goals; contrast across schools.

Parsing into Semantic Graphs (1:00)

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

Outlook: Using Semantic Graphs (0:15)

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

Outline: Our Game Plan



Foundations: Linguistic & Formal (0:30)

► Tease apart various 'facets' (layers) of meaning; common terminology.

Graph-Based Meaning Banks (0:45)

- ► Semi-superficial review of five English corpora with semantic graphs;
- ▶ highlight distinct design decisions and goals; contrast across schools.

Parsing into Semantic Graphs (1:00)

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

Outlook: Using Semantic Graphs (0:15)

 \blacktriangleright Example use cases: summarization, entity linking, machine translation.

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

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.

7

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

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

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

7

Abrams gave Browne a book.

Abrams gave a book to Browne.

Browne was given a book by Abrams.

A book was given to Browne by Abrams.

Browne, Abrams gave the book to.

A book, Browne was given by Abrams.

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

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

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



Abrams gave Browne a book.

Abrams gave a book to Browne.

Browne was given a book by Abrams.

A book was given to Browne by Abrams.

Browne, Abrams gave the book to.

A book, Browne was given by Abrams.

. . .

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

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

- ► Superficially different linguistic forms can describe the same situation;
- ▶ hold true under the same circumstances; can substitute for each other;
- → close paraphrases: convey the 'same meaning' (in unmarked contexts).



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.

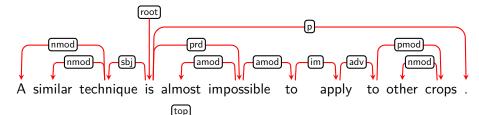
Han elsker sin hatt. He_i loves his_i hat. Han elsker hans hatt. He_i loves his_{$i\neq i$} hat.

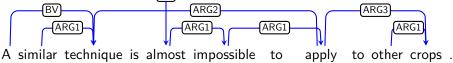


Sherlock saw the suspect with the binoculars.



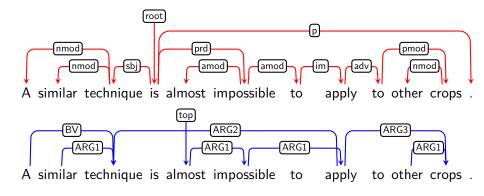
Syntactic Trees vs. Semantic Graphs





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

Syntactic Trees vs. Semantic Graphs



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

Different Desiderata (and Levels of Abstraction)

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

9

Semi-Formally: Trees vs. Graphs



Structural Wellformedness Conditions on Trees

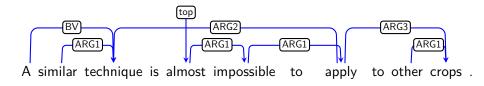
- ▶ Unique root, connected, single parent, free of cycles; maybe: projective;
- ightarrow all nodes (but the root) reachable by unique directed path from root.

Semi-Formally: Trees vs. Graphs



Structural Wellformedness Conditions on Trees

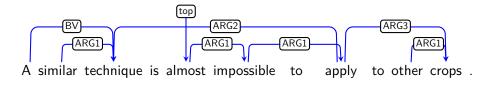
- ▶ Unique root, connected, single parent, free of cycles; maybe: projective;
- \rightarrow all nodes (but the root) reachable by unique directed path from root.



Semi-Formally: Trees vs. Graphs

Structural Wellformedness Conditions on Trees

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



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).

Terminology: Syntactic vs. Semantic 'Predicates'



Syntax: Head–Dependent Relations

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

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

Terminology: Syntactic vs. Semantic 'Predicates'



Syntax: Head–Dependent Relations

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

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



Syntax: Head–Dependent Relations

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

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

Semantics: Predicate-Argument Relations

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



Syntax: Head–Dependent Relations

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

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

Semantics: Predicate-Argument Relations

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



Syntax: Head–Dependent Relations

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

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

Semantics: Predicate-Argument Relations

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



Syntax: Head–Dependent Relations

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

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

Semantics: Predicate-Argument Relations

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



Syntax: Head-Dependent Relations

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

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

Semantics: Predicate-Argument Relations

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

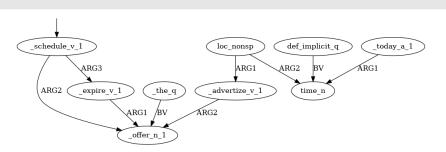
Abrams bet \$10 on Monday that it rained in Florence.

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

Reflections on Predicate–Argument Structure

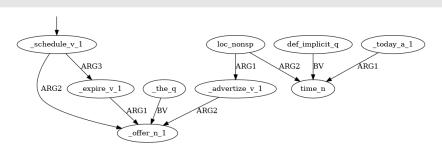


Arguments recursively are predicates most of the time (and vice versa);



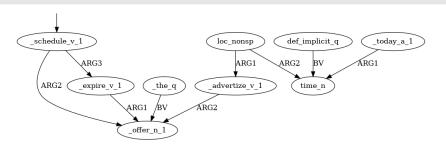
Reflections on Predicate–Argument Structure

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



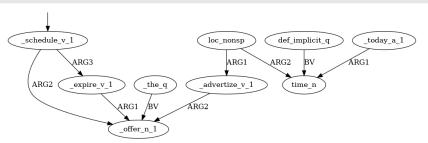
Reflections on Predicate–Argument Structure

- 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): offer' $(x) \land \text{schedule'}(_, x, \text{expire'}(x))$



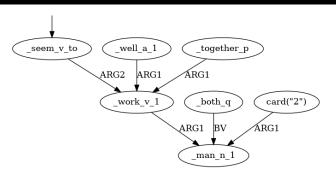
Reflections on Predicate-Argument Structure

- 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): offer' $(x) \land schedule'(_, x, expire'(x))$
- decomposition: possibly multiple predicates per word or construction.



Mismatches between Syntax and Semantics



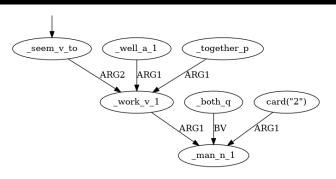


Both men seem to work well together. [WSJ #0109043]

- 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

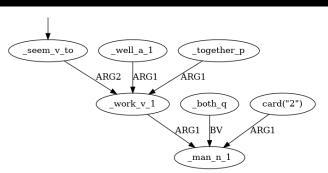




Both men seem to work well together. [WSJ #0109043]

- 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



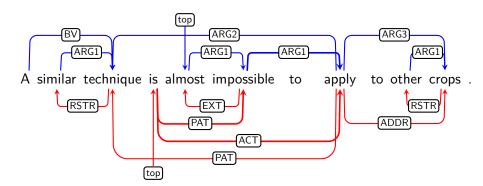
Both men seem to work well together. [WSJ#0109043]

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

Example Design Decisions: Copula Constructions

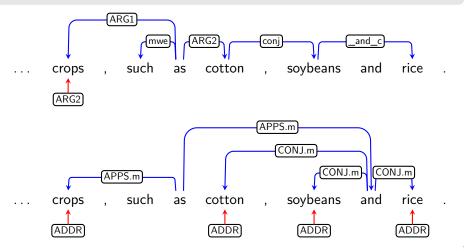


- ► There is much variation in analysis of individual linguistic phenomena;
- specific semantic framework requires many interacting design decisions;
- divergent views already at the level of which words are content-bearing;
- ▶ for example, the predicative copula: the fierce dog vs. the dog is fierce.



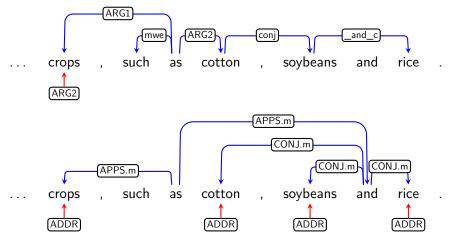
Example Design Decisions: Coordination

- -
- ▶ 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



Same Coordinate Structure in EDS, AMR, Maybe UCCA

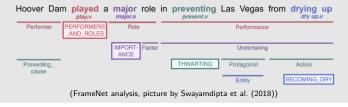
Popel et al. (2013) Survey Syntactic Dependencies

Partial Predicate-Argument Structures in SRL



SRL: Semantic Role Labeling

- ► Represent predicate—argument structure with focus on verbal predicates.
- ► Large, manually annotated corpora available (for multiple languages), e.g. PropBank (Palmer et al., 2005), Framenet (Baker et al., 1998).



Not trivial to view as integrated, 'full-sentence' meaning representation:

- ▶ Internal structure of spans: edge from *play* to *prevent* or *dry up*?
- ► No constraint that SRL graph should be connected.
- ► Many content words and semantic phenomena remain unannotated.

History: Before Semantic Graphbanks, e.g. FraCaS



Fracas #248

- P1 The PC-6082 is faster than any ITEL computer.
- P2 The ITEL-ZX is an ITEL computer.
- Q Is the PC-6082 faster than the ITEL-ZX?
- H The PC-6082 is faster than the ITEL-ZX.
- ► 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.

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.

Model Theory

- ► Many semantic representation formalisms (e.g. predicate logic, DRT) admit a model-theoretic interpretation:
 - ▶ syntax: the semantic representation itself, e.g. a logical formula
 - \blacktriangleright $\underline{\mathsf{model}} :$ formal representation of a possible world, e.g. in terms of sets
 - ► <u>interpretation:</u> under what exact conditions will a formula be <u>true</u> in a model?
- Many semantic representation formalisms (in particular, all graph-based ones) do not have model-theoretic interpretations.
- Without model theory, it is hard to model things like quantification.

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

Presupposition and Focus



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

Presupposition

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

Focus

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

Word Senses



The words in a sentence may be ambiguous with respect to their <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.

- ► "small investor"
- ▶ represent as small → investor?
- ▶ represent as (p / person :arg0-of (i / invest-01 :manner (s
 / small)))? as AMR graph

In Sum, Facets of Linguistic Meaning Predicate-Argument Structure Sense Differentiation Scope and Quantification Anaphoric Coreference Presupposition and Focus Grounding ► e.g. Wikipedia linking

Compositionality











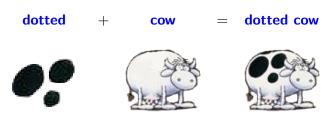
(Example by Jan van Eijck)

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





(Example by Jan van Eijck)

The Principle of Compositionality

The meaning of every expression in a language must be a function of the meaning of its immediate constituents and the syntactic rule used to combine them.

A. Goldberg

Compositionality





(Example by Jan van Eijck)

The Principle of Compositionality

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

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;
- ightharpoonup a node n is reentrant if $\operatorname{\textit{in}}(n) > 1$ (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;
- $ightharpoonup T \subseteq N$ 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);
- 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;

Graph Theory 101



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

- ▶ 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 and Edge Decorations



Predicates vs. Constants

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

Graph Structure vs. Node and Edge Decorations



Predicates vs. Constants

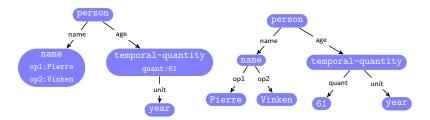
- ► Nodes and edges can be labeled (e.g. by relation and role identifiers);
- ▶ labels can be internally structured: node properties and edge attributes;
- properties (and attributes) are non-recursive attribute-value matrices;
- ▶ node (and edge) label is merely a distinguished property (or attribute);

Graph Structure vs. Node and Edge Decorations



Predicates vs. Constants

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





- ► 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; can impact evaluation.

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

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 Meaning Banks

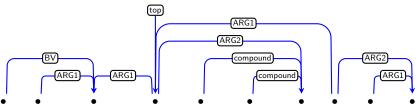
- ▶ PropBank (Palmer et al., 2005), Framenet (Baker et al., 1998), ...;
- ► Groningen Parallel Meaning Bank: GMB, PMB (Basile et al., 2012);
- ► Universal Decompositional Semantics (White et al., 2016);
- oniversal Decompositional Semantics (vville et al., 2010),



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

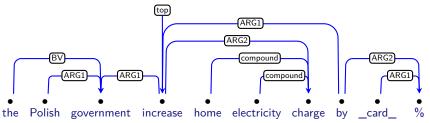


- ► Two decades of great advances in syntactic dependencies and parsing;
- ► recently, renewed interest in meaning; algorithmic interest in graphs;
- ▶ nodes limited to surface lexical units (words):



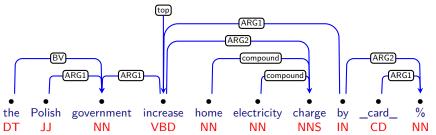


- ► Two decades of great advances in syntactic dependencies and parsing;
- recently, renewed interest in meaning; algorithmic interest in graphs;
- ▶ nodes limited to surface lexical units (words): lemmas



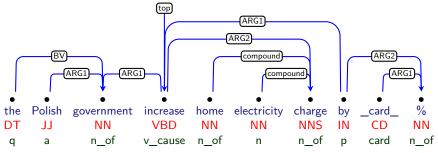


- ► Two decades of great advances in syntactic dependencies and parsing;
- ► recently, renewed interest in meaning; algorithmic interest in graphs;
- ▶ nodes limited to surface lexical units (words): lemmas, PoS



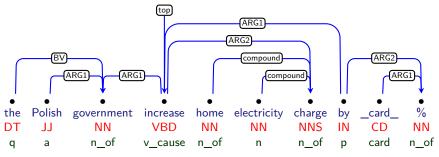


- ► Two decades of great advances in syntactic dependencies and parsing;
- recently, renewed interest in meaning; algorithmic interest in graphs;
- ▶ nodes limited to surface lexical units (words): lemmas, PoS, and frames;



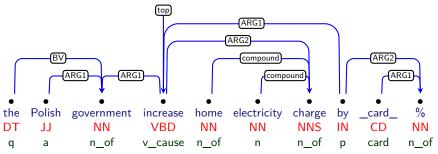


- ► Two decades of great advances in syntactic dependencies and parsing;
- ► recently, renewed interest in meaning; algorithmic interest in graphs;
- ▶ nodes limited to surface lexical units (words): lemmas, PoS, and frames;
- edges encode argument roles and maybe some construction semantics;





- ► 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.





Background

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



Background

- ► Parallel work on bi-lexical 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 bi-lexical digraphs;
- ▶ 34,004 + 1348 sentences, 745,543 + 29,808 tokens from WSJ corpus;
- ▶ three frameworks: DM, PAS, PSD; nine participating teams: 78–92 F_1 .



Background

- ▶ Parallel work on bi-lexical 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 bi-lexical digraphs;
- ▶ 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);
- + frame (or sense) prediction; evaluation beyond 'atomic' dependency $\mathsf{F}_1.$

Background

- Parallel work on bi-lexical 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 bi-lexical digraphs;
- ▶ 34,004 + 1348 sentences, 745,543 + 29,808 tokens from WSJ corpus;
- ▶ three frameworks: DM, PAS, PSD; nine participating teams: 78–92 F₁.

+ Out-of-domain English test data from Brown Corpus: 1,849; 31,583;

SemEval 2015 (Oepen et al., 2015)

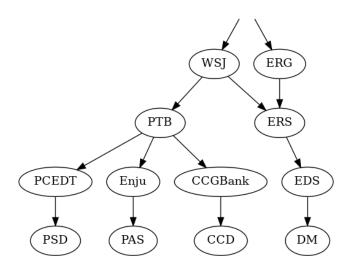
- + Chinese PAS (32,783; 687,433) and Czech PSD (48,972; 1,111,626);
- + frame (or sense) prediction; evaluation beyond 'atomic' dependency F₁.

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

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

Bilexical Semantic Dependencies: Genealogy

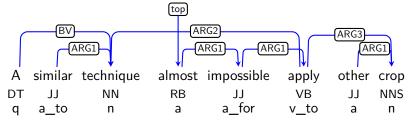




DELPH-IN MRS Bi-Lexical Dependencies (DM)



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



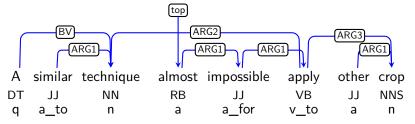
Ivanova et al. (2013)

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



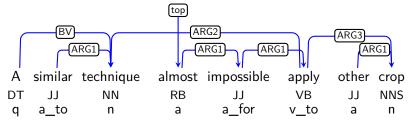
Ivanova et al. (2013)

- Simplification from underspecified logical forms (ERS; coming right up);
- ightharpoonup edges 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. (2013)

- ► Simplification from underspecified logical forms (ERS; coming right up);
- lacktriangle edges 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)

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



LinGO English Resource Grammar (Flickinger, 2000)

- ► Hand-designed computational grammar for English in HPSG framework;
- declarative, unification-based: parsing and realization; multiple engines;
- ▶ 20⁺ 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)

- ► Hand-designed computational grammar for English in HPSG framework;
- ▶ declarative, unification-based: parsing and realization; multiple engines;
- ▶ 20⁺ 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)

- ► Hand-designed computational grammar for English in HPSG framework;
- declarative, unification-based: parsing and realization; multiple engines;
- ▶ 20⁺ 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)

- ► Hand-designed computational grammar for English in HPSG framework;
- declarative, unification-based: parsing and realization; multiple engines;
- ▶ 20⁺ 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};



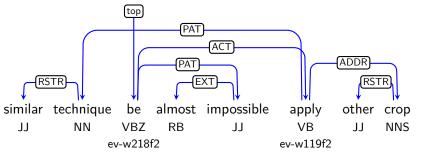
LinGO English Resource Grammar (Flickinger, 2000)

- ► Hand-designed computational grammar for English in HPSG framework;
- ▶ declarative, unification-based: parsing and realization; multiple engines;
- $\blacktriangleright~20^+$ 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};
- ► record full HPSG derivation; export into various graph-based formats.

Prague Semantic Dependencies (PSD)

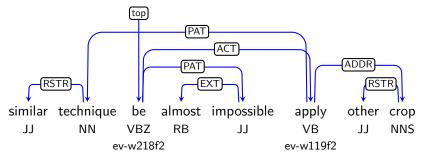
A similar technique is almost impossible to apply to other crops.



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

Prague Semantic Dependencies (PSD)

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

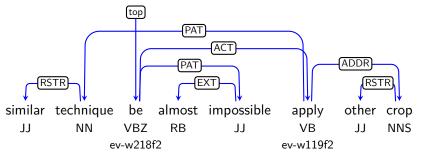


- Simplification of multi-layer dependency annotation (coming right up);
- ► ACT(or), PAT(ient), ADDR(essee), ORIG(in), EFF(ect): complements;
- ▶ first verbal argument is ACT, also in causative—inchoative alternation;

Prague Semantic Dependencies (PSD)



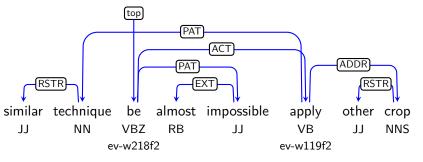
A similar technique is almost impossible to apply to other crops.



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

Prague Semantic Dependencies (PSD)

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



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

Background: Functional Generative Description



Linguistic Theory (Sgall et al., 1986)

- ► Charles University in Prague, since 1960s; (Peter 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; (Peter 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: 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.

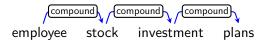
pedt A PAT similar technique PAT is almost impossible 0 * to apply *T*-1 to ADDR other crops ADDR, such as cotton ADDR, soybeans ADDR and rice ADDR.

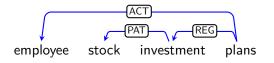
pedt Armstrong expects *-1 to close the sale of the color unit in late
November and the carpet sale in December, with the PAT gains PAT to
be applied *-4 to ADDR fourth quarter or first-quarter results ADDR.

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?







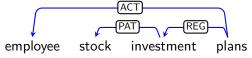
Diverging Ambitions

- ▶ 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?



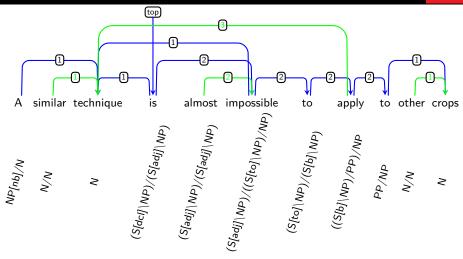




Diverging Ambitions

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

Other Bilexical 'Semantic' Dependencies (1 of 2)

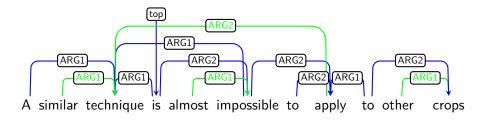


CCD: CCG Word–Word Dependencies (Hockenmaier & Steedman, 2007)

- ► CCG categories as 'frame' identifiers; edge labels for argument position;
- more 'deep syntax' than semantics, but functor—argument directionality.

Other Bilexical 'Semantic' Dependencies (2 of 2)



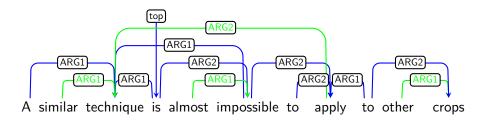


PAS: Enju Predicate–Argument Structures (Miyao, 2006)

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

Other Bilexical 'Semantic' Dependencies (2 of 2)



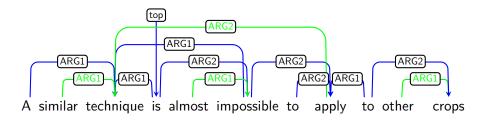


PAS: Enju Predicate–Argument Structures (Miyao, 2006)

- 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 (2 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.

Pairwise Similarity (Unlabeled Dependency F₁)



	Directed			U		
	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			U	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		U	ndirected	
	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.

Limitations in Bi-Lexical Semantic Dependencies



Lexical Decomposition

Construction Semantics

Sub-Lexical Meaning

Elementary Dependency Structures (EDS)



Construction Semantics (in EDS vs. DM)



A Few More Reflections on EDS



Radical Compositionality

Sense Differentiation

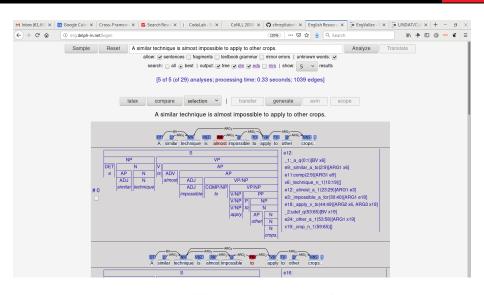
Limitations

Dependency Minimal Recursion Semantics (DMRS)



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)

Predicate-Argument Structure in AMR

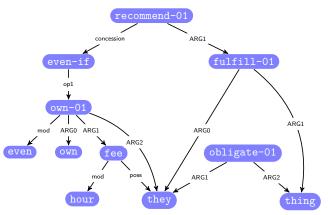




The boy wants to sleep.

More Complex AMRs





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

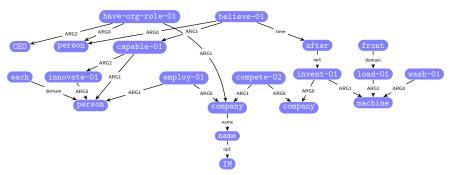
(a) implicit arguments

(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 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.

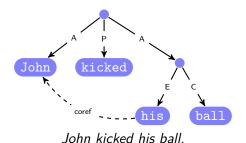
► Basic ideas:

- ▶ Backbone of UCCA graph is a tree with the tokens as leaves and additional internal nodes, connected by a small set (12) of semantic relations.
- ► Additional edges represent argument sharing.
- ► Multiple annotation layers, e.g. p-a 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)

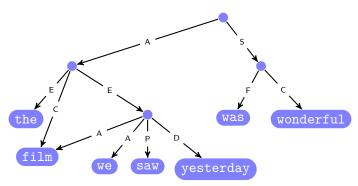
UCCA Terminology

- ► 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

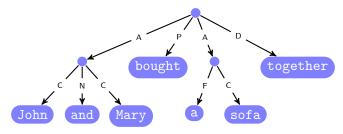


UCCA: Argument sharing



The film we saw yesterday was wonderful.

UCCA: Coordination



John and Mary bought a sofa together.

Limitations of UCCA



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

Graphbank Statistics

			CCD	DM	PSD	EDS	AMR	AMR-
S	(01)	number of graphs	39604	35656	35656	35656	10309	10309
COUNTS	(02)	average number of tokens	23.47	22.51	22.51	22.51	20.62	20.62
5	(03)	average number of nodes per token	0.88	0.77	0.64	0.99	0.67	0.6
O	(04)	number of edge labels	6	59	90	10	135	100
	(05)	%g trees	1.45	2.31	42.26	0.98	52.48	18.6
	(06)	%g treewidth one	29.27	69.82	43.08	65.37	52.72	52.7
	(07)	average treewidth	1.742	1.303	1.614	1.352	1.524	1.52
IKEENESS	(08)	maximal treewidth	5	3	7	3	4	
Ž	(09)	average edge density	1.070	1.019	1.073	1.047	1.065	1.06
Ξ.	(10)	% _n reentrant	28.09	27.43	11.41	28.42	5.23	18.9
Ξ	(11)	%g cyclic	1.28	0.00	0.00	0.04	3.15	0.7
	(12)	%g not connected	12.53	6.57	0.70	1.49	0.00	0.0
	(13)	%g multi-rooted	99.67	99.49	99.33	98.75	0.00	77.5
	(14)	percentage of non-top roots	47.78	44.94	4.34	41.15	0.00	19.3
Ę	(15)	average edge length	2.582	2.684	3.320	-	-	
OKDEK	(16)	%g noncrossing	48.23	69.21	64.61	-	-	
5	(17)	$%_g$ pagenumber two	98.64	99.55	98.07	-	-	
	(01)	number of graphs	87	87	87	87	87	8
	(03)	average number of nodes per token	0.88	0.79	0.64	1.01	0.66	0.6
	(05)	%g trees	1.15	1.15	45.98	1.15	60.92	3.4
5	(06)	% treewidth one	37.93	81.61	47.13	81.61	60.92	60.9
CONTROL	(07)	average treewidth	1.644	1.184	1.540	1.184	1.402	1.40
Z	(09)	average edge density	1.057	1.011	1.061	1.028	1.038	1.03
5	(10)	$%_n$ reentrant	28.92	27.73	10.28	27.77	2.88	21.0
	(11)	%g cyclic	0.00	0.00	0.00	0.00	2.30	0.0
	(12)	%g not connected	6.90	3.45	1.15	1.15	0.00	0.0
	(13)	%g multi-rooted	100.00	100.00	100.00	98.85	0.00	93.1

(Kuhlmann & Oepen, 2016)

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.
- Semi-automatically annotated with Discourse Representation Structures (DRT).

Facets of Meaning (Revisited)



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

• • •

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



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



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



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	unan chored	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



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

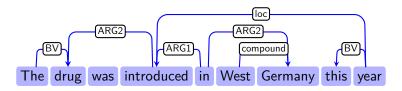
The drug was introduced in West Germany this year



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	unan chored	AMR	no sub-string correspondences annotated

Parsing to Flavor (0) graphs

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



Semantic Parsing is Making Rapid Progress



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

Accuracy of edge prediction

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

Semantic Parsing is Making Rapid Progress



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

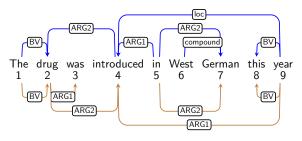
Quantifying graph similarity is challenging. What do these numbers mean?



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

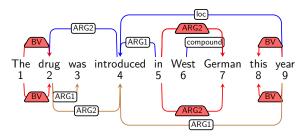
$$|E_{\mathsf{gold}}| = 7$$





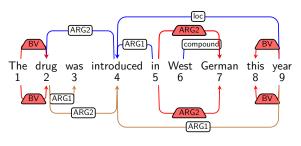
$$\begin{split} E_{\mathsf{gold}} &= \{(1,2,\mathsf{bv}), (2,4,\mathsf{arg1}), \cdots\} \\ E_{\mathsf{system}} &= \{(1,2,\mathsf{bv}), (2,3,\mathsf{arg1}), \cdots\} \end{split} \qquad \begin{aligned} |E_{\mathsf{gold}}| &= 7 \\ |E_{\mathsf{system}}| &= 6 \end{aligned}$$





```
\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}
```

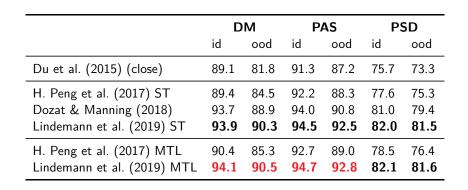




$$\begin{split} E_{\mathsf{gold}} &= \{ (1, 2, \mathsf{bv}), (2, 4, \mathsf{arg1}), \cdots \} & |E_{\mathsf{gold}}| = 7 \\ E_{\mathsf{system}} &= \{ (1, 2, \mathsf{bv}), (2, 3, \mathsf{arg1}), \cdots \} & |E_{\mathsf{system}}| = 6 \\ E_{\mathsf{match}} &= E_{\mathsf{gold}} \cap E_{\mathsf{system}} =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

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

Magic Numbers Again



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

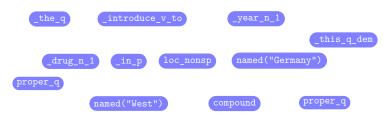
Parsing to Flavor (1) and Flavor (2) Graphs



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



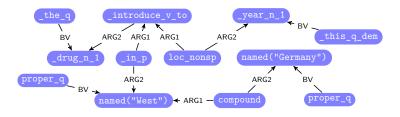
Parsing to Flavor (1) and Flavor (2) Graphs



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



	EDS	5	AMR 2015	AMR 2017
	Smatch F	EDM	Smatch F	Smatch F
Groschwitz et al. (2018)	-	-	70.2	71.0
Lyu & Titov (2018)	-	-	73.7	74.4
Zhang et al. (2019)	-	-	-	76.3
Buys & Blunsom (2017)	85.5	85.9	60.1	_
Chen et al. (2018)	90.9	90.4	-	_
Lindemann et al. (2019)	90.1	84.9	74.3	75.3
+ MTL	90.4	85.2	74.5	75.3

Accuracy of node and edge prediction

Semantic Parsing is Making Rapid Progress



	EDS	5	AMR 2015	AMR 2017
	Smatch F	EDM	Smatch F	Smatch F
Groschwitz et al. (2018)	-	-	70.2	71.0
Lyu & Titov (2018)	-	-	73.7	74.4
Zhang et al. (2019)	-	-	-	76.3
Buys & Blunsom (2017)	85.5	85.9	60.1	-
Chen et al. (2018)	90.9	90.4	-	-
Lindemann et al. (2019)	90.1	84.9	74.3	75.3
+ MTL	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 $_5$. $_6$

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

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

 $boy_n_1\langle 1,2\rangle$

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



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

_the_q $\langle 0,1
angle$

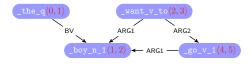
 $\boxed{\texttt{_want_v_to}\langle 2,3\rangle}$

 $boy_n_1\langle 1,2\rangle$

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

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





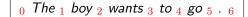
$$V_{\mathsf{gold}} = \{(\langle 0, 1 \rangle\,, _\mathsf{the_q}), \cdots \}, \, |V_{\mathsf{gold}}| = 4$$





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

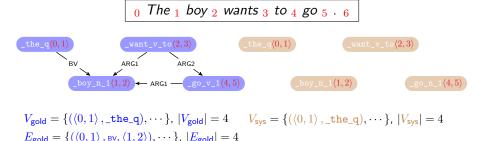




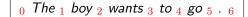


$$\begin{split} V_{\mathrm{gold}} &= \{ (\langle 0, 1 \rangle\,,_\mathtt{the_q}), \cdots \}, \ |V_{\mathrm{gold}}| = 4 \\ E_{\mathrm{gold}} &= \{ (\langle 0, 1 \rangle\,, \mathsf{BV}, \langle 1, 2 \rangle), \cdots \}, \ |E_{\mathrm{gold}}| = 4 \end{split}$$











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





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



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

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



$$\begin{array}{c} \text{ the_q(0,1)} \\ \text{BV} \\ \text{ARG1} \\ \text{ARG2} \\ \text{BV} \\ \text{ARG1} \\ \text{ARG2} \\ \text{BV} \\ \text{ARG1} \\ \text{BV} \\ \text{ARG1} \\ \text{ARG2} \\ \text{BV} \\ \text{ARG1} \\ \text{BV} \\ \text{ARG1} \\ \text{Cov}_n_1\langle 1,2 \rangle \longleftrightarrow \text{ARG1} \longrightarrow \text{Cov}_n_1\langle 4,5 \rangle \\ \text{Cov}_n_1\langle 1,2 \rangle \longleftrightarrow \text{ARG1} \longrightarrow \text{Cov}_n_1\langle 4,5 \rangle \\ \text{Cov}_n_1\langle 1,2 \rangle \longleftrightarrow \text{ARG1} \longrightarrow \text{Cov}_n_1\langle 4,5 \rangle \\ \text{Cov}_n_1\langle 1,2 \rangle \longleftrightarrow \text{ARG2} \\ \text{Cov}_n_1\langle 1,2 \rangle \longleftrightarrow \text{ARG3} \longrightarrow \text{Cov}_n_1\langle 1,2 \rangle \longleftrightarrow \text{ARG4} \\ \text{Cov}_n_1\langle 1,2 \rangle \longleftrightarrow \text{ARG4} \longrightarrow \text{Cov}_n_1\langle 1,2 \rangle \longleftrightarrow \text{Cov}_n_1\langle 1$$

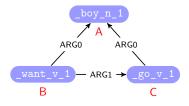
$$\begin{split} &V_{\mathsf{gold}} = \{(\langle 0, 1 \rangle\,,_\mathsf{the_q}), \cdots\}, \ |V_{\mathsf{gold}}| = 4 & V_{\mathsf{sys}} = \{(\langle 0, 1 \rangle\,,_\mathsf{the_q}), \cdots\}, \ |V_{\mathsf{sys}}| = 4 \\ &E_{\mathsf{gold}} = \{(\langle 0, 1 \rangle\,,\mathsf{BV}, \langle 1, 2 \rangle),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

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

$$\begin{aligned} & \mathsf{EDM}_P & \mathsf{EDM}_A & \mathsf{EDM} \\ & \frac{2*|V_{\mathsf{match}}|}{|V_{\mathsf{gold}}|+|V_{\mathsf{sys}}|} = 0.86 & \frac{2*|E_{\mathsf{match}}|}{|E_{\mathsf{gold}}|+|E_{\mathsf{sys}}|} = 0.57 & \frac{2*(|V_{\mathsf{match}}|+|E_{\mathsf{match}}|)}{|V_{\mathsf{gold}}|+|V_{\mathsf{sys}}|+|E_{\mathsf{gold}}|+|E_{\mathsf{sys}}|} = 0.67 \end{aligned}$$

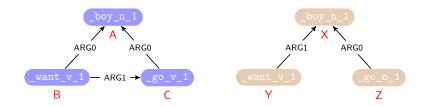


Flavor	Name	Example	Type of Anchoring
1	anchored	EDS	free node–sub-string correspondences
2	unanchored	AMR	no sub-string correspondences annotated



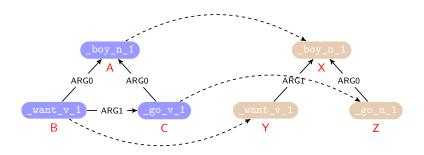


Flavor	Name	Example	Type of Anchoring
1	anchored	EDS	free node–sub-string correspondences
2	unanchored	AMR	no sub-string correspondences annotated





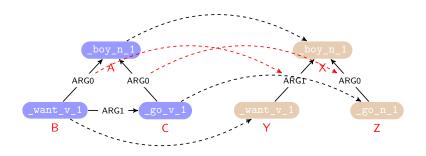
Flavor	Name	Example	Type of Anchoring
1	anchored	EDS	free node-sub-string correspondences
2	unanchored	AMR	no sub-string correspondences annotated



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



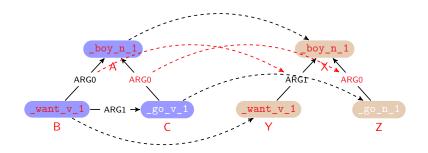
Flavor	Name	Example	Type of Anchoring
1	anchored	EDS	free node-sub-string correspondences
2	unanchored	AMR	no sub-string correspondences annotated



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



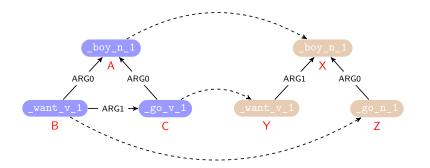
Flavor	Name	Example	Type of Anchoring
1	anchored	EDS	free node–sub-string correspondences
2	unanchored	AMR	no sub-string correspondences annotated



EDM_P	EDM_A	EDM	
$\frac{2*2}{3+3} = 0.67$	$\frac{2*1}{3+2} = 0.40$	$\frac{2*3}{6+5} = 0.55$	



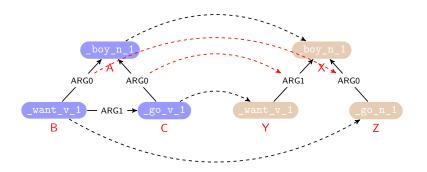
Flavor	Name	Example	Type of Anchoring
1	anchored	EDS	free node–sub-string correspondences
2	unanchored	AMR	no sub-string correspondences annotated



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



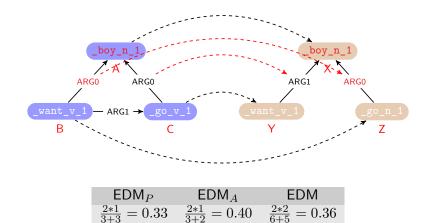
Flavor	Name	Example	Type of Anchoring
1	anchored	EDS	free node–sub-string correspondences
2	unanchored	AMR	no sub-string correspondences annotated



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



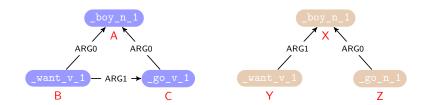
Flavor	Name	Example	Type of Anchoring
1	anchored	EDS	free node–sub-string correspondences
2	unanchored	AMR	no sub-string correspondences annotated



Evaluation for Parsing to Flavor (2) Graphs



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



$$\frac{\mathsf{SMATCH}(G_g, G_s)}{\mathsf{SMATCH}(G_g, G_s)} = \max_{a \in \mathcal{A}(G_g, G_s)} \mathsf{EDM}(a)$$

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

Magic Numbers Again

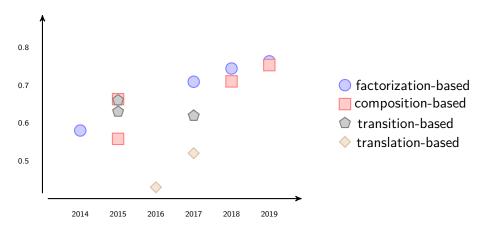


		AMR 2015 Smatch F	AMR 2017 Smatch F
-	-	70.2	71.0
-	-	73.7	74.4
-	-	-	76.3
85.5	85.9	60.1	-
90.9	90.4	-	-
90.1	84.9	74.3	75.3
90.4	85.2	74.5	75.3
	Smatch F 85.5 90.9 90.1		Smatch F EDM Smatch F - - 70.2 - - 73.7 - - - 85.5 85.9 60.1 90.9 90.4 - 90.1 84.9 74.3

Modern graph parsers are cool!

Graph Parsing is Making Rapid Progress



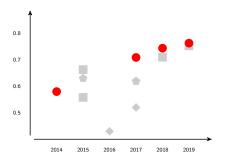


AMR parsing accuracies reported in Flanigan et al. (2014); Foland & Martin (2017); Lyu & Titov (2018); Zhang et al. (2019);

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



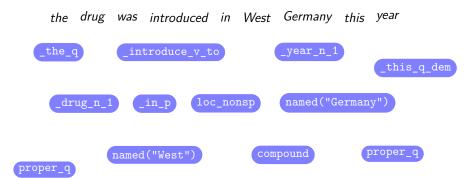


- ► Inspired by the successful design of graph-based dependency parsers.
- ► Very early work: McDonald & Pereira (2006).
- ► Explicitly modeling the target structure.
- ► A parser knows how to evaluate the *goodness* of a candidate graph.
- ► A parser knows how to find the *best* graph from an extremely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

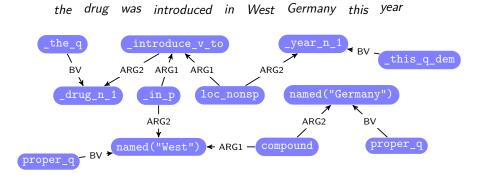


the drug was introduced in West Germany this year







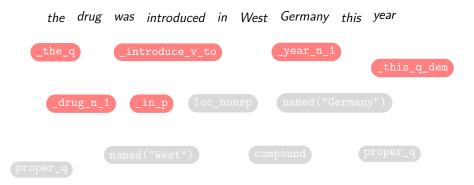






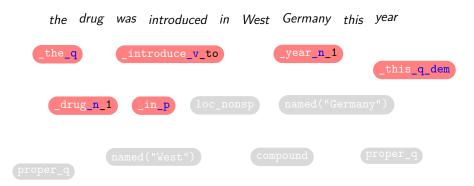
Task 1: Concept Identification





Task 1: Concept Identification





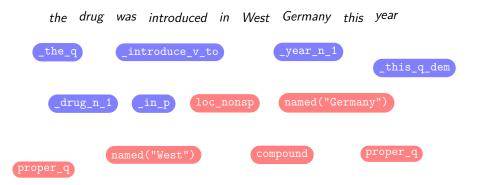
Task 1: Concept Identification





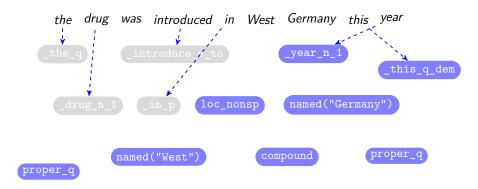
Task 1: Concept Identification





Task 1: Concept Identification

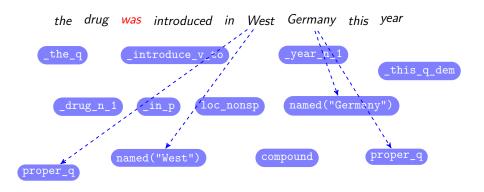




Task 0: Concept-to-word Alignment

Task 1: Concept Identification

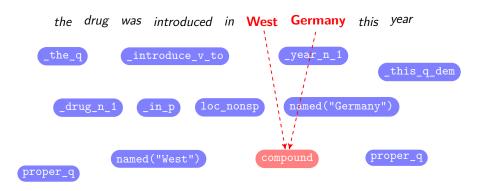




Task 0: Concept-to-word Alignment

Task 1: Concept Identification

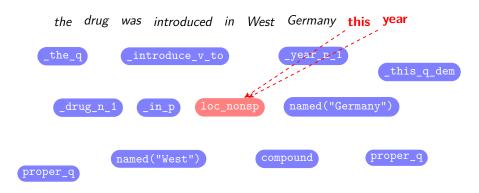




Task 0: Concept-to-word Alignment

Task 1: Concept Identification





Task 0: Concept-to-word Alignment

Task 1: Concept Identification





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

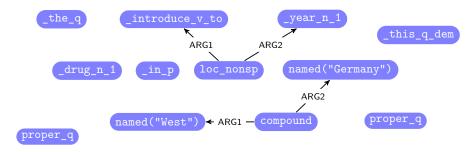




- Task 0: Concept-to-word Alignment
- Task 1: Concept Identification
- Task 2: Relation Detection



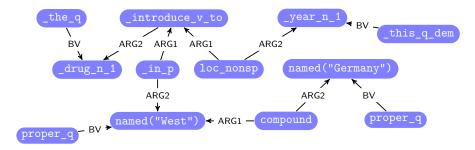
the drug was introduced in West Germany this year



- 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

Sub-Tasks

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

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

Sub-Tasks

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

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

As a Structured Prediction Problem



Maximum Subgraph Parsing

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

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

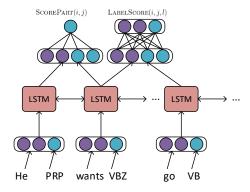
First-order factoriztion

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

Marco Kuhlmann and Peter Jonsson. 2015. Parsing to Noncrossing Dependency Graphs.

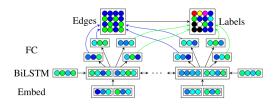


$$G' = \arg\max_{G^* = (V, E^* \subseteq E)} \sum_{e \in E^*} \text{ScorePart}(e)$$



Hao Peng, Sam Thomson and Noah Smith. 2017. Deep Multitask Learning for Semantic Dependency Parsing.





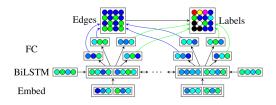
Concatenate word and POS tag embeddings.

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

► Then *BiLSTM* them:

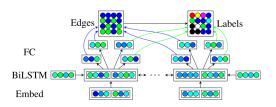
$$\mathbf{r}_1, \mathbf{r}_2, ..., \mathbf{r}_n = \mathsf{BiLSTM}(\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_n)$$



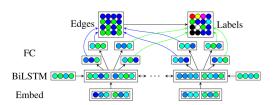


- ► A word can be either a predicate or an argument.
- ► Distinguish its grammartical/semantic function:

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



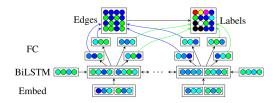
$$\begin{array}{lcl} \text{ScorePart}(s,i,j) & = & \text{biaffine}^{\text{edge}}(\mathbf{h}_i^{\text{edge-head}},\mathbf{h}_i^{\text{edge-dep}}) \\ \mathbf{h}_i^{\text{edge-head}} & = & \text{fnn}^{\text{edge-head}}(\mathbf{r}_i) \\ \mathbf{h}_i^{\text{edge-dep}} & = & \text{fnn}^{\text{edge-dep}}(\mathbf{r}_i) \\ & \text{fnn}(\mathbf{x}) & = & \text{ReLU}(\mathbf{W}\mathbf{x} + \mathbf{b}) \end{array}$$



Calculate label scores:

$$\begin{aligned} \mathbf{h}_i^{\mathsf{label-head}} &= {_{\mathrm{FNN}}}^{\mathsf{label-head}}(\mathbf{r}_i) \\ \mathbf{h}_i^{\mathsf{label-dep}} &= {_{\mathrm{FNN}}}^{\mathsf{label-dep}}(\mathbf{r}_i) \\ \\ \mathrm{SCORELABEL}^{\mathsf{label}}(i,j) &= \mathrm{BIAFFINE}^{\mathsf{label}}(\mathbf{h}_i^{\mathsf{label-head}}, \mathbf{h}_i^{\mathsf{label-dep}}) \end{aligned}$$





- ► If ScorePart $(s, i, j) \ge 0$, then take $i \to j$ as an edge.
- ▶ Select a label for $i \to j$ according to SCORELABEL label (i, j)

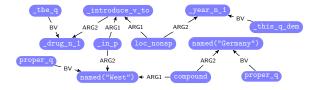
Sub-Tasks

- ► 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)	✓	✓	✓

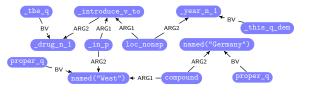


Flavor	Name	Example	Type of Anchoring
1	anchored	EDS	free node-sub-string correspondences
2	unanchored	AMR	no sub-string correspondences annotated





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 d	rug was	introduced	in	West	Germany	this	year
_the_q	ıg_n_1 ∅	introduce_v_to	_in_p	named("W")	named("G") proper_q	_this_q_dem	_year_n_1
					pound	loc_n	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	1	_in_p	named("W") proper_q	named("G") proper_q	_this_q_dem	_year_n_1
						compound			loc_nonsp

Almost Sequence Labeling

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

- ► 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	_in_p		named("W") proper_q		named("G") proper_q	this_q_dem		_year_n_1
					Ī	com	po	und	loc_n	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
 - ightharpoonup I-x: inside x
- ► Lightweight phrase-structure parsing (UCCA)

Neural Tagging

Almost Sequence Labeling

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

Challenge

Like POS tagging but with thousands of labels.

Delexicalization

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

Neural Tagging



Almost Sequence Labeling

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

Challenge

Like POS tagging but with thousands of labels.

Delexicalization

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



Solution

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





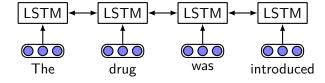






Solution

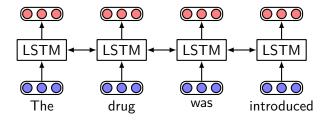
- ► Pretraining: ELMo, BERT, etc.
- ► Word encoder: LSTM, Transformer, etc.
- Classification





Solution

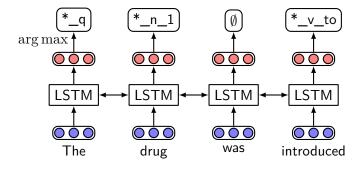
- ► Pretraining: ELMo, BERT, etc.
- ► Word encoder: LSTM, Transformer, etc.
- Classification





Solution

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



Sub-Tasks

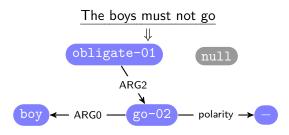
- ► Concept Identification (CI): predicting nodes
- ► Relation Detection (RD): linking nodes
- ► Concept-to-word alignment: finding concept-word correspondences

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

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

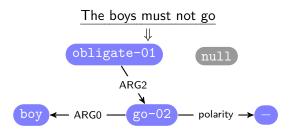


AMR Parsing





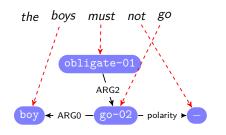
AMR Parsing



No Annotations for Concept-to-Word Alignment

- ► Heuristic rules (JAMR)
- ► Linearize graphs and reuse word alignment tools, e.g. giza++ and BerkeleyAligner, etc.
- Consider all possible alignments





Reference

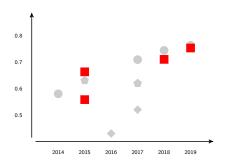
Chunchuan Lyu, Ivan Titov. 2018. AMR Parsing as Graph Prediction with Latent Alignment

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

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

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 *grammar*.

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

Reminder: Compositionality





(Example from Jan van Eijck)

The Principle of Compositionality

the meaning of every expression in a language must be a function of the meaning of its immediate constituents and the syntactic rule used to combine them.

A. Goldberg

Reminder: Compositionality











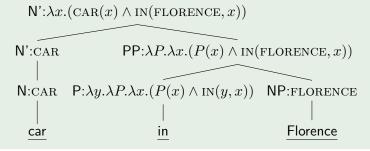
(Example from Jan van Eijck)

The Principle of Compositionality

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

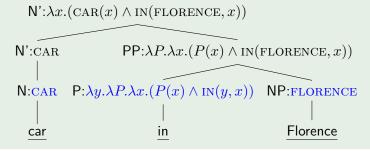


Tree construction + Lexical interpretation \Rightarrow Meaning representation



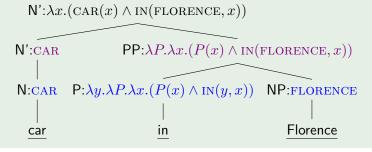


Tree construction + Lexical interpretation \Rightarrow Meaning representation



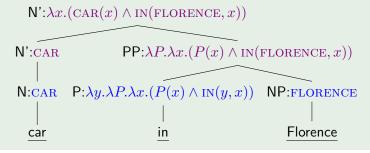


Tree construction + Lexical interpretation \Rightarrow Meaning representation





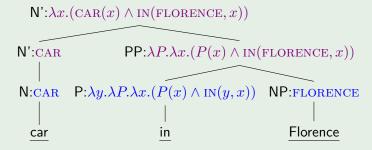
Tree construction + Lexical interpretation \Rightarrow Meaning representation





Tree construction + Lexical interpretation \Rightarrow Meaning representation

Using λ Expressions



Manipulating Graphs

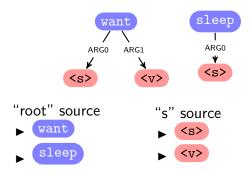
- ► Graph algebra (AM algebra)
- Graph grammar (hyperedge replacement grammar)



S-Graph

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

Examples





The writer wants to sleep soundly

AM algebra provides a systematic way to construct graphs

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

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



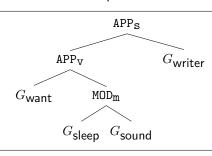
The writer wants to sleep soundly

 \Downarrow

AM algebra provides a systematic way to construct graphs

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

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



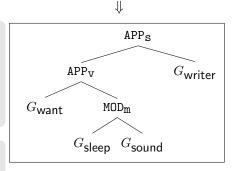


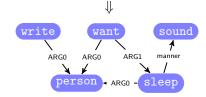
The writer wants to sleep soundly

AM algebra provides a systematic way to construct graphs

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

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





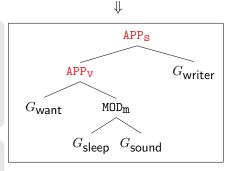


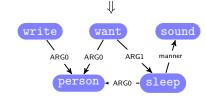
The writer wants to sleep soundly

AM algebra provides a systematic way to construct graphs

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

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





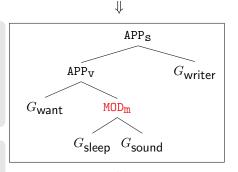


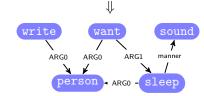
The writer wants to sleep soundly

AM algebra provides a systematic way to construct graphs

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

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

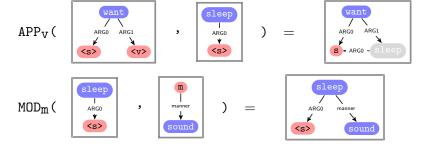




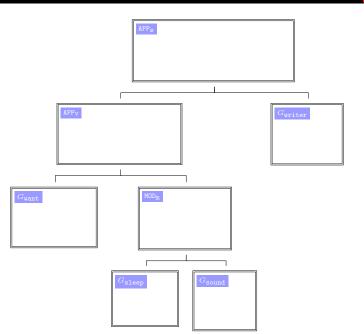


Two operations for combining s-graphs:

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

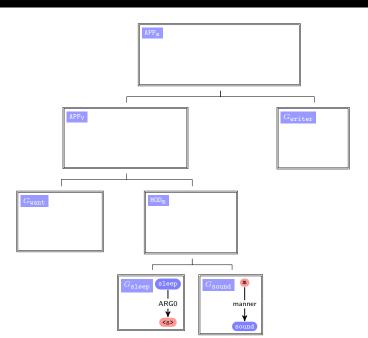


Jonas Groschwitz, Meaghan Fowlie, Mark Johnson and Alexander Koller. 2017. A constrained graph algebra for semantic parsing with AMRs.

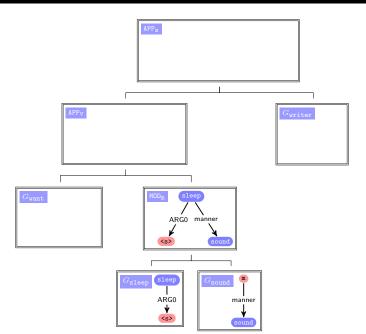




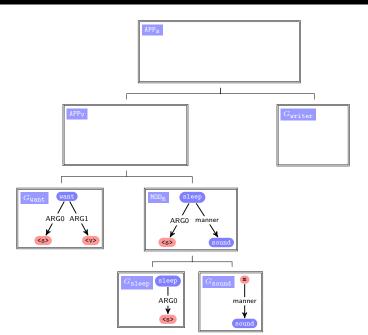




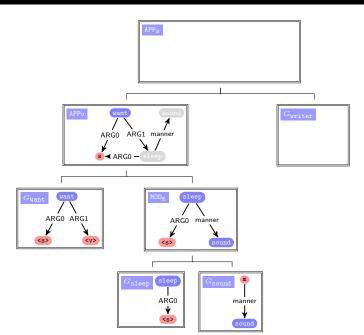




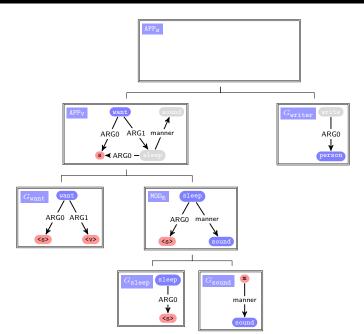


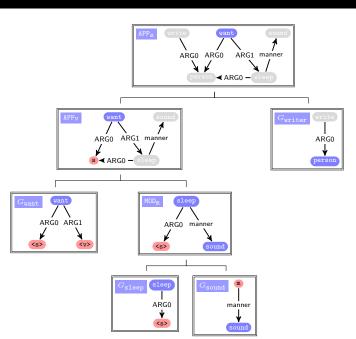










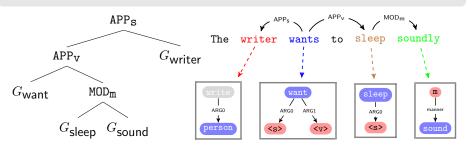




AM Algebra-Based Parsing



Tree construction + Lexical interpretation ⇒ Meaning representation



Well-typed AM term with alignments of leaves to tokens in string can be represented as a (semantic!) dependency tree.

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.

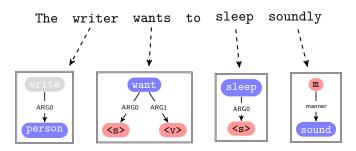
The writer wants to sleep soundly

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.

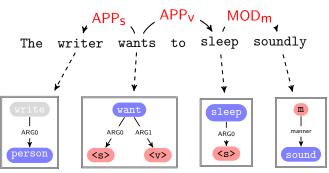


AM Algebra-Based Parsing



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.

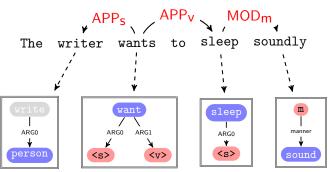


AM Algebra-Based Parsing



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.

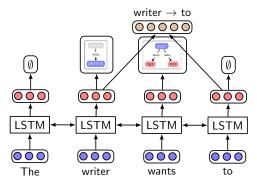


AM Algebra-Based Parsing



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.



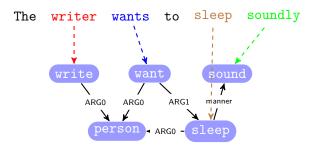


The writer wants to sleep soundly



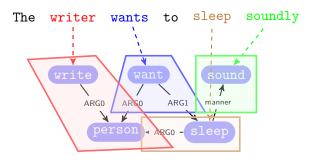
Decompose graph into parts based on concept-to-word alignment.





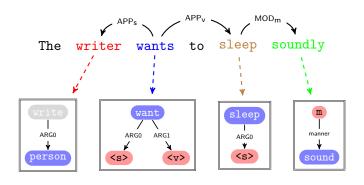
Decompose graph into parts based on concept-to-word alignment.





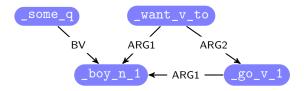
Decompose graph into parts based on concept-to-word alignment.





Training data = supertags + AM dependency tree

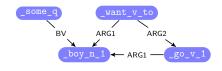
Hypergraph

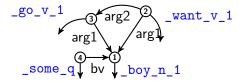


A graph consists of:

- ► 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 .



$$\mathring{\downarrow} S \stackrel{\gamma_1}{\Longrightarrow} {\text{arg1}} \bigvee_{\textbf{NP}} {\text{NP}}$$

- ► 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 .



$$\label{eq:state_state} \bigvee \mathbf{S} \ \stackrel{\gamma_1}{\Longrightarrow} \ \underset{\mathbf{NP}}{\operatorname{arg1}} \ \stackrel{\mathbf{VP}}{\Longrightarrow} \ \underset{\mathbf{DET}}{\operatorname{bv}} \ \underset{\mathbf{NNS}}{\operatorname{prg1}}$$

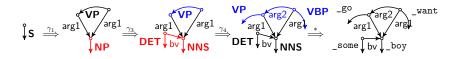
- Terminal vs. non-terminal hyperedges
- ► Non-terminal hyperedges are utilized to control a derivation process.
- ► A derivation starts from a non-terminal hyperedge .
- ► In a derivation step, we substitute a non-terminal hyperedge with a hypergraph .





- ► 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.



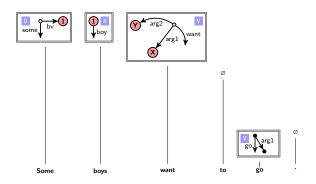


- ► Terminal vs. non-terminal hyperedges (symbols)
- ► Non-terminal hyperedges (symbols) are utilized to control a derivation process.
- ► A derivation starts from a non-terminal hyperedge (symbol).
- ► In a derivation step, we substitute a non-terminal hyperedge (symbols) with a hypergraph (a sequence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



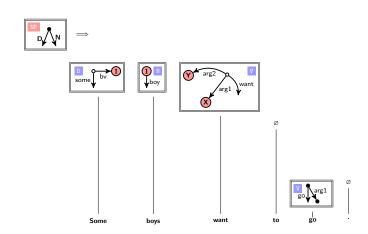
Some boys want to go .

105

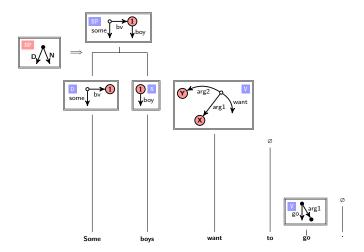




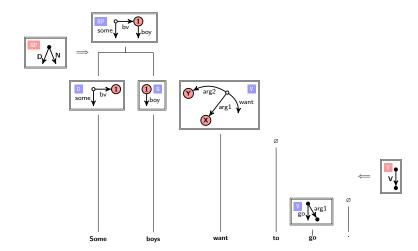




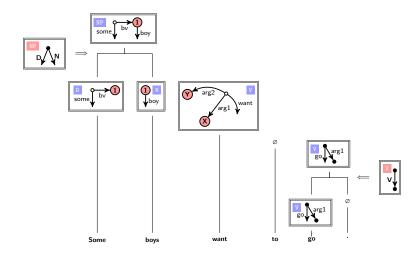




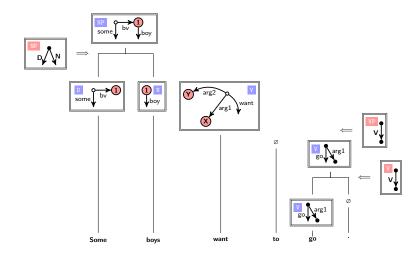




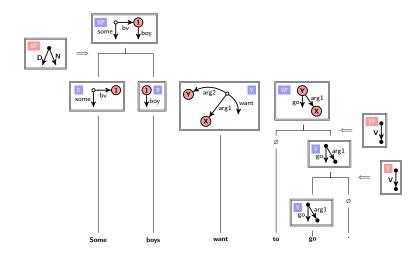




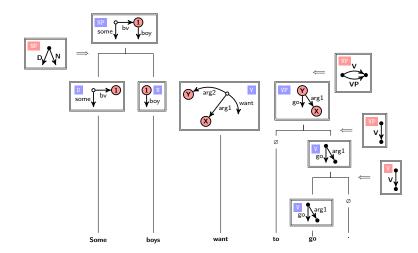




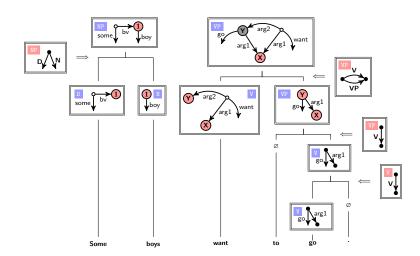




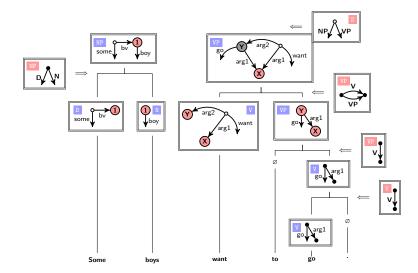




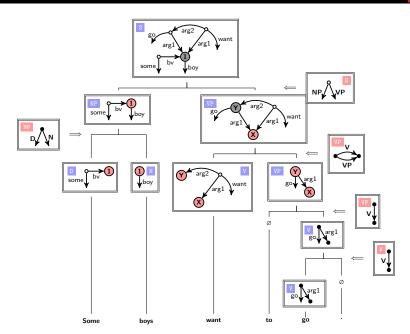










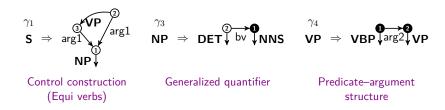


HRGs Can be Linguistically Meaningful

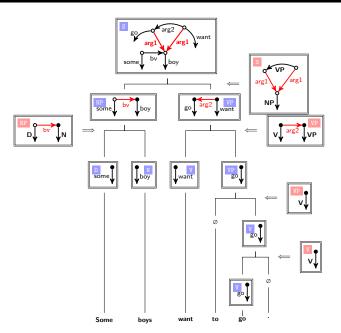




Rules



Construction Semantics (Revisited)



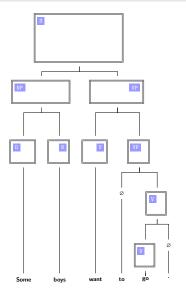




Tree construction + Semantic interpretation ⇒ Meaning representation

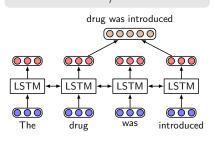


Tree construction + Semantic interpretation ⇒ Meaning representation



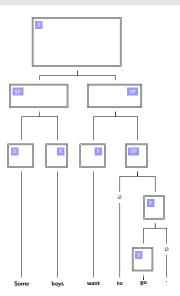
Syntactic parsing

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



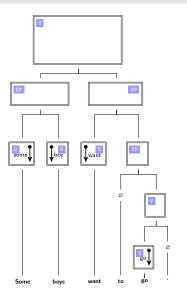


Tree construction + Semantic interpretation \Rightarrow Meaning representation





Tree construction + Semantic interpretation ⇒ Meaning representation

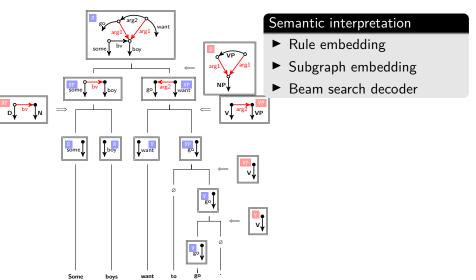


Semantic interpretation

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



Tree construction + Semantic interpretation ⇒ Meaning representation



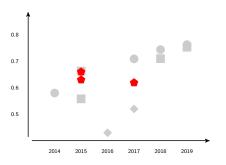
Magic Numbers



	DM		PAS		PSD		EDS		AMR	
	id	ood	id	ood	id	ood	Smatch	EDM	2015	2017
Groschwitz et al. (2018)	-	-	-	-	-	-	-	-	70.2	71.0
Lyu & Titov (2018)	-	-	-	-	-	-	-	-	73.7	74.4
Zhang et al. (2019)	-	-	-	-	-	-	-	-	-	76.3
H. Peng et al. (2017)	89.4	84.5	92.2	88.3	77.6	75.3	-	-	-	-
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 et al. (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
H. Peng et al. (2017)	90.4	85.3	92.7	89.0	78.5	76.4	-	-	-	-
Lindemann et al. (2019)	94.1	90.5	94.7	92.8	82.1	81.6	90.4	85.2	74.5	75.3

Transition-Based Approaches





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

Transition-Based Parsing



- A transition system for parsing is a quadruple $S = (C, T, c_s, C_t)$, where
- 1. C is a set of configurations, each of which represents a parser state.
- 2. T is a set of transitions, each of which represents a parsing action,
- 3. c_s initializes S by mapping a sentence x to a particular configuration,
- 4. $C_t \subseteq C$ is a set of terminal configurations.

Deterministic parsing

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

1 $c \leftarrow c_s(x)$
2 while $c \notin C_t$
3 $c = \text{Act}(c, \text{GetTransition}(c))$
4 return G_c



	Some 1	-			_
U	1	2	3	7	J

Transition

- ► Shift
- ► Left-Arc
- ► Right-Arc
- ▶ Delete

Stack [TOP] Buffer

[Some, boys, want, to, go]

Transition Shift



TOP	Some	boys	want	to	go
0	1	2	3	4	5

Transition

► Shift

► Left-Arc

► Right-Arc

► Delete

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

Transition



TOP	Some	boys	want	to	go
0	1	2	3	4	5

Transition

► Shift

► Left-Arc

► Right-Arc

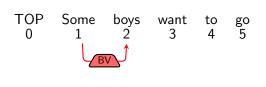
▶ Delete

Stack [TOP, Some]

Buffer [boys, want, to, go]

Transition
Right-Arc(BV)





Transition

► Shift

► Left-Arc

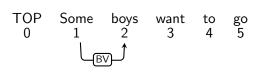
► Right-Arc

▶ Delete

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

Transition





Transition

► Shift

► Left-Arc

► Right-Arc

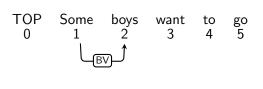
► Delete

Stack [TOP, Some]

Buffer [boys, want, to, go]

Transition Delete





Transition

► Shift

► Left-Arc

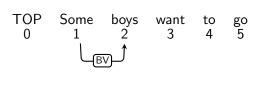
► Right-Arc

▶ Delete

Stack [TOP] Buffer [boys, want, to, go]

Transition





Transition

► Shift

► Left-Arc

► Right-Arc

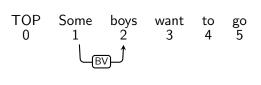
▶ Delete

Stack [TOP] Buffer

[boys, want, to, go]

Transition Shift





Transition

► Shift

► Left-Arc

► Right-Arc

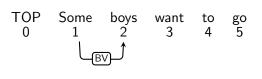
▶ Delete

Stack [TOP, boys]

Buffer [want, to, go]

Transition





Transition

► Shift

► Left-Arc

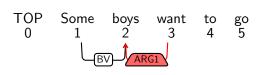
► Right-Arc

▶ Delete

Stack [TOP, boys] Buffer [want, to, go]

Transition
Left-Arc(ARG1)





Transition

► Shift

► Left-Arc

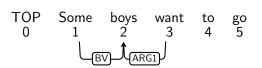
► Right-Arc

▶ Delete

Stack [TOP, boys] Buffer [want, to, go]

Transition





Transition

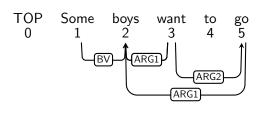
- ► Shift
- ► Left-Arc
- ► Right-Arc
- ► Delete

Stack

Buffer

Transition





Transition

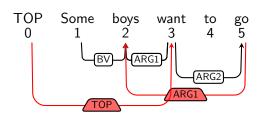
- ► Shift
- ► Left-Arc
- ► Right-Arc
- ► Delete

Stack

Buffer

Transition





Transition

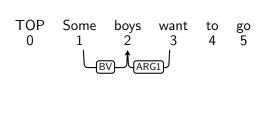
- ► Shift
- ► Left-Arc
- ► Right-Arc
- ▶ Delete

Stack

Buffer

Transition





Transition

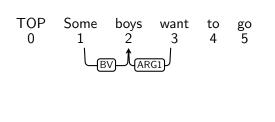
- ► Shift
- ► Left-Arc
- ► Right-Arc
- Delete
- Swap

Stack [TOP, boys]

Buffer [want, to, go]

Transition Swap





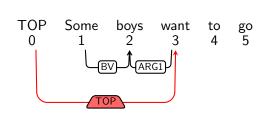
Transition

- ► Shift
- ► Left-Arc
- ► Right-Arc
- Delete
- Swap

Stack [boys, TOP] Buffer [want, to, go]

Transition





Transition

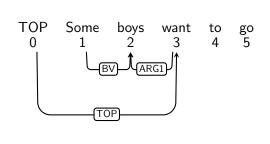
- ► Shift
- ► Left-Arc
- ► Right-Arc
- ► Delete
- Swap

Stack [boys, TOP]

Buffer [want, to, go]

Transition
Right-Arc(TOP)





Transition

► Shift

▶ Left-Arc▶ Right-Arc

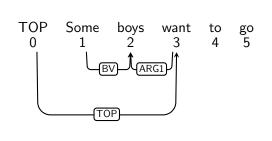
► Delete

► Swap

Stack [boys, TOP] Buffer [want, to, go]

Transition





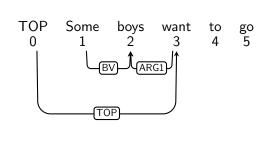
Transition

- ► Shift
- ► Left-Arc
- ► Right-Arc
- ► Delete
- Swap

Stack [boys, TOP] Buffer [want, to, go]

Transition Delete





Transition

► Shift

► Left-Arc

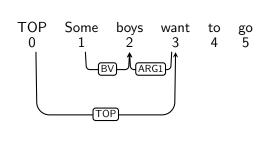
Right-ArcDelete

► Swap

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

Transition





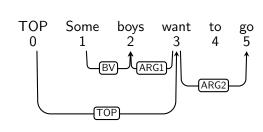
Transition

- ► Shift
- ► Left-Arc
- ► Right-Arc
- ► Delete
- Swap

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

Transition





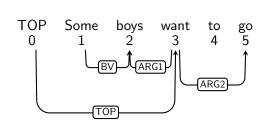
Transition

- ► Shift
- ► Left-Arc
- ► Right-Arc
- ▶ Delete
- Swap

Stack [boys] Buffer [go]

Transition





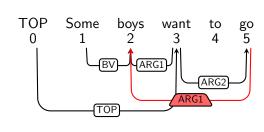
Transition

- ► Shift
- ► Left-Arc
- ► Right-Arc
- Delete
- Swap

Stack [boys] Buffer [go]

Transition
Left-Arc(ARG1)





Transition

- ► Shift
- ► Left-Arc
- ► Right-Arc
- Delete
- Swap

Stack

Buffer

Transition

Challenges



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

► Parameterized Arc actions

► Parameterized Shift actions

Challenges



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

► Parameterized Arc actions

► Parameterized Shift actions

Challenges



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

► Parameterized Arc actions

► Parameterized Shift actions



Transition

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

Stack

Buffer

[Some, boys, want, to, go]

Transition Init(_some_q)



```
_some_q
```

Transition

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

Stack

Buffer

 $[_{\tt some_q}, \ boys, \ want, \ to, \ go]$

Transition



```
_some_q
```

Transition

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

```
Stack
```

Buffer

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

```
Transition Shift(_boy_n_1)
```



```
_some_q
```

_boy_n_1

Transition

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

Stack [_some_q]

Buffer

 $[_boy_n_1$, want, to, go]

Transition



```
_some_q
```

_boy_n_1

Transition

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

Stack [_some_q]

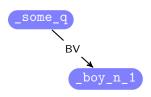
Transition

Buffer

[_boy_n_1, want, to, go]

Right-Arc(BV)





Transition

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

Stack [_some_q]

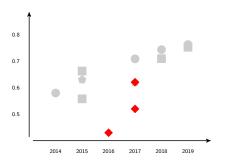
Buffer

[_boy_n_1, want, to, go]

Transition

Translation-Based Approaches

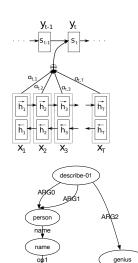




- ► Inspired by neural machine translation.
- Semantic graph as a foreign language.
- ► A parser knows how to linearize a graph.
- ► Data augmentation has been shown very helpful, partially relecting the data-hungry nature of seq2seq models.

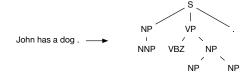
Sequence-to-Sequence Models for Parsing





"Ryan"

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

Integrated with Other Approaches



$Translation + Transition-Based \ Approach$

Xiaochang Peng, Linfeng Song, Daniel Gildea and Giorgio Satta. 2018. Sequence-to-sequence Models for Cache Transition Systems.

- ► 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

$Translation + Factorization - Based\ Approach$

Sheng Zhang, Xutai Ma, Kevin Duh and Benjamin Van Durme. 2019. AMR Parsing as Sequence-to-Graph Transduction.

- ▶ Predicting nodes with a sequence-to-sequence model
- ► Linking nodes with a factorization model

Cross-Framework Parsing



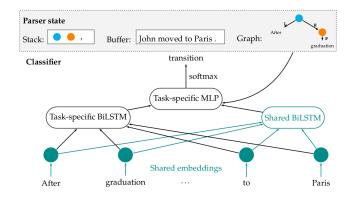
- ► State-of-the-art graph parsers rely on large-scale, manually annotated graphbanks.
- ► Multiple heterogeneous graphbanks EXIST!
- ?> Can we get a better parser by integrating heterogeneous graphbanks?
- 1. Heterogeneous annotations are (similar but) different.
 - ► Different projects, different linguistic 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

Hal Daumé III. 2007. Frustratingly easy domain adaptation.



Daniel Hershcovich, Omri Abend and Ari Rappoport. 2018. Multitask Parsing Across Semantic Representations

Magic Numbers



Improving a	Transition-Based	Parser	for	UCCA
-------------	------------------	--------	-----	------

	In-Do	main	Out-of-Domain		
	Primary Remote		Primary	Remote	
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 an AM Algebra–Based Parser for Many Graphbanks

	D	OM PAS			PSD id ood		EDS EDM	AMR 2015 Smatch
Single (GloVe) Single (BERT)	90.4	84.3	91.4	86.6	78.1	74.5	82.5	69.2 74.3
MTL (GloVe) MTL (BERT)				88.0 92.8			83.3 85.2	70.4 74.5

Magic Numbers



Improving a Transition-Based Parser for UCCA

	In-Do	main	Out-of-Domain		
	Primary Remote		Primary	Remote	
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 an AM Algebra-Based Parser for Many Graphbanks

		M .			PSD		EDS	AMR 2015
	id	ood	id	ood	id	ood	EDM	Smatch
Single (GloVe) Single (BERT)								69.2 74.3
MTL (GloVe) MTL (BERT)					78.9 82.1		83.3 85.2	70.4 74.5

Cross-lingual parsing

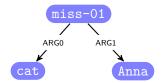


Multilingual Parsing

- ► 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

- lacktriangle 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



1. shore up a decline 支撑……减少的局面/situation [wsj#20012010]

2. four-color page 四色印刷页面

[wsj#20012005]

3. was for dinner and dancing 是……晚餐舞会

[wsj#20010016]

4. from the same period last year 同比

[wsj#20011007] >we have a single word

5. were particularly dusty/ADJ 灰尘/NOUN尤其大/ADJ [wsj#20003025]



1. shore up a decline 支撑……减少的局面/situation [wsj#20012010]

2. four-color page 四色印刷页面

[wsj#20012005]

3. was for dinner and dancing 是……晚餐舞会

[wsj#20010016]

4. from the same period last year 同比

[wsj#20011007] >we have a single word

5. were particularly dusty/ADJ 灰尘/NOUN尤其大/ADJ [wsj#20003025]



1. shore up a decline 支撑……减少的局面/situation [wsj#20012010]

2. four-color page 四色印刷页面

[wsi#20012005]

3. was for dinner and dancing 是……晚餐舞会

>coordination ⊳compound

> [wsj#20010016] [wsi#20011007]

>we have a single word

[wsi#20003025]

4. from the same period last year 同比

5. were particularly dusty/ADJ 灰尘/NOUN尤其大/ADJ



1. shore up a decline 支撑……减少的局面/situation

[wsj#20012010]

2. four-color page 四色印刷页面

[wsj#20012005]

3. was for dinner and dancing 是……晚餐舞会 [wsj#20010016]

4. from the same period last year 同比

[wsj#20011007]

5. were particularly dusty/ADJ 灰尘/NOUN尤其大/ADJ ⊳we have a single word

[wsj#20003025]



1. shore up a decline 支撑……减少的局面/situation [wsj#20012010]

2. four-color page 四色印刷页面

[wsj#20012005]

3. was for dinner and dancing 是……晚餐舞会

[wsj#20010016]

4. from the same period last year 同比

[wsj#20011007] >we have a single word

we have a single

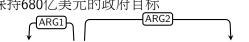
[wsj#20003025]

5. were particularly dusty/ADJ 灰尘/NOUN尤其大/ADJ



[wsi#20011005]

1. remain under a government target of \$68 billion 保持680亿美元的政府目标

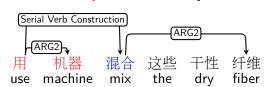


remain under a government target of \$68 billion



2. mechanically 用机器

mechanically mixed the dry fibers



[wsj#20003026]

Cross-Lingual Parsing



Challenges

- ► Factorization-based approach: alignment
- ► Composition-based approach: cross-lingual compositionality is rather weak.

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

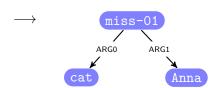
DE: Anna fehlt ihrem Kater

-

- 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.

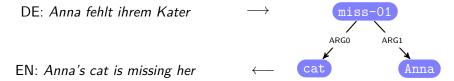
Example

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).



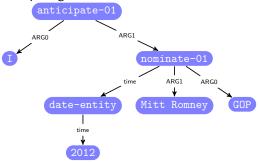
Task: Map entities in text to entities in a knowledge base (KB).

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 ...

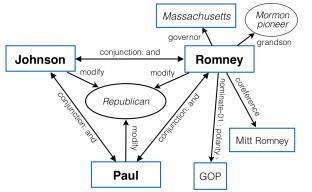


Task: Map entities in text to entities in a knowledge base (KB).

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 ...



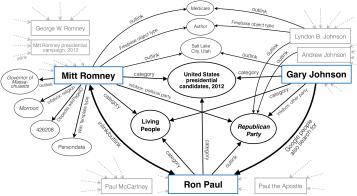


Task: Map entities in text to entities in a knowledge base (KB).

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

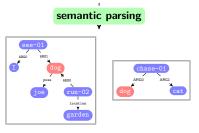
Pan et al. (2015)



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

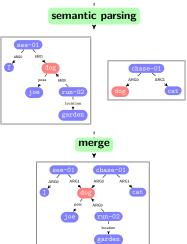


I saw Joe's dog, which was running in the garden.





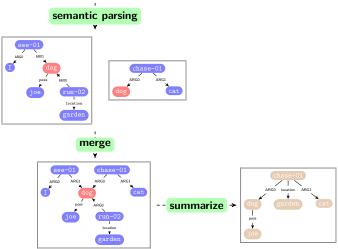
I saw Joe's dog, which was running in the garden.



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



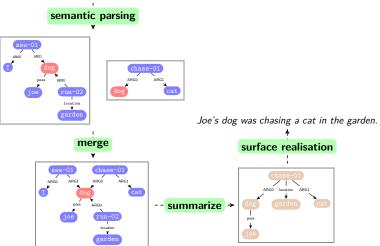
I saw Joe's dog, which was running in the garden.



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



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



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.
- Generalize across graphbanks?
- ► Check out CoNLL 2019 Shared Task on Cross-Framework Meaning Representation Parsing (MRP, http://mrp.nlpl.eu/).

Facets of Meaning

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

Acknowledgments



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 eacl*.
- 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.
- 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.

References II

- 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: Association for Computational Linguistics.
- 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.

References III

- 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.
- 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.

References IV

- 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.
- 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* (pp. 484–490). Melbourne, Australia.
- 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* (semeval 2015) (pp. 927–931). Denver, Colorado: Association for Computational Linguistics.

References V

- 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.
- Foland, W., & Martin, J. H. (2017). Abstract meaning representation parsing using LSTM recurrent neural networks. In *Proceedings of the 55th annual meeting of the association for computational linguistics* (volume 1: Long papers) (pp. 463–472). Vancouver, Canada: Association for Computational Linguistics.
- 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.

References VI

- 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 emnlp.*
- Hockenmaier, J., & Steedman, M. (2007). CCGbank. A corpus of CCG derivations and dependency structures extracted from the Penn Treebank. *Computational Linguistics*, *33*, 355–396.
- Ivanova, A., Oepen, S., Dridan, R., Flickinger, D., & Øvrelid, L. (2013). On different approaches to syntactic analysis into bi-lexical dependencies. An empirical comparison of direct, PCFG-based, and HPSG-based parsers. In *Proceedings of the 13th International Conference on Parsing Technologies* (pp. 63–72). Nara, Japan.

References VII

- Jones, B., Andreas, J., Bauer, D., Hermann, K. M., & Knight, K. (2012). Semantics-based machine translation with hyperedge replacement grammars. In *Proceedings of coling*.
- 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., & 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*.

References VIII

- Liu, F., Flanigan, J., Thomson, S., Sadeh, N., & Smith, N. A. (2015). Toward abstractive summarization using semantic representations. In *Proceedings of naacl.*
- Lyu, C., & Titov, I. (2018). AMR parsing as graph prediction with latent alignment. In *Proceedings of the 56th annual meeting of the association for computational linguistics (volume 1: Long papers)* (pp. 397–407). Melbourne, Australia: Association for Computational Linguistics.
- 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.

References IX

- Oepen, S., Flickinger, D., Toutanova, K., & Manning, C. D. (2004). LinGO Redwoods. A rich and dynamic treebank for HPSG. *Research on Language and Computation*, 2(4), 575–596.
- Oepen, S., Kuhlmann, M., Miyao, Y., Zeman, D., Cinková, S., Flickinger, D., ... Urešová, Z. (2015). SemEval 2015 Task 18. Broad-coverage semantic dependency parsing. In *Proceedings of the 9th International Workshop on Semantic Evaluation* (pp. 915–926). Denver, CO, USA.
- Oepen, S., Kuhlmann, M., Miyao, Y., Zeman, D., Flickinger, D., Hajič, J., ... Zhang, Y. (2014). SemEval 2014 Task 8. Broad-coverage semantic dependency parsing. In *Proceedings of the 8th International Workshop on Semantic Evaluation* (pp. 63–72). Dublin, Ireland.
- Palmer, M., Gildea, D., & Kingsbury, P. (2005). The Proposition Bank. A corpus annotated with semantic roles. *Computational Linguistics*, 31(1), 71-106.

References X

- Pan, X., Cassidy, T., Hermjakob, U., Ji, H., & Knight, K. (2015). Unsupervised entity linking with abstract meaning representation. In *Proceedings of human language technologies: The 2015 annual conference of the north american chapter of the acl.*
- 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.

References XI

- Peng, X., Wang, C., Gildea, D., & Xue, N. (2017). Addressing the data sparsity issue in neural AMR parsing. In *Proceedings of the 15th conference of the European chapter of the association for computational linguistics: Volume 1, long papers* (pp. 366–375). Valencia, Spain: Association for Computational Linguistics.
- Reddy, S., Lapata, M., & Steedman, M. (2014). Large-scale semantic parsing without question-answer pairs. *Transactions of the ACL*.
- 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.
- 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.
- Swayamdipta, S., Thomson, S., Lee, K., Zettlemoyer, L., Dyer, C., & Smith, N. A. (2018). Syntactic scaffolds for semantic structures. In *Proceedings of emnlp.*

References XII

- Wang, C., Xue, N., & Pradhan, S. (2015a). Boosting transition-based AMR parsing with refined actions and auxiliary analyzers. In *Proceedings* of the 53rd annual meeting of the association for computational linguistics and the 7th international joint conference on natural language processing (volume 2: Short papers) (pp. 857–862). Beijing, China: Association for Computational Linguistics.
- 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: Human language technologies* (pp. 366–375). Denver, Colorado: Association for Computational Linguistics.
- 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.

References XIII

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*.