MRP 2019:
Cross-Framework Meaning Representation Parsing

http://mrp.nlpl.eu

Stephan Oepen♣, Omri Abend♠, Jan Hajič♥,
Daniel Hershcovitch♦, Marco Kuhlmann◦,
Tim O’Gorman*, and Nianwen Xue•

♣ University of Oslo, Department of Informatics
♠ The Hebrew University of Jerusalem, School of Computer Science and Engineering
♥ Charles University in Prague, Institute of Formal and Applied Linguistics
♦ University of Copenhagen, Department of Computer Science
◦ Linköping University, Department of Computer and Information Science
* University of Colorado at Boulder, Department of Linguistics
• Brandeis University, Department of Computer Science

mrp-organizers@nlpl.eu
A similar technique is almost impossible to apply to other crops.
similar technique be almost impossible apply other crop
JJ NN VBZ RB JJ VB JJ NNS
10,000-Meter Perspective: Parsing into Semantic Graphs
a similar technique almost impossible apply other crop
q a_to technique a for impossible a_for v_to a
DT JJ NN RB JJ NNS
Almost a 1

Impossible a for

Arg1

Technique n 1

Similar a to

Arg1

Apply v to

Arg1

Crop n 1

Udef q

Other a 1
Why Graph-Based Meaning Representation?

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

The dog was chasing a cat.
Why Graph-Based Meaning Representation?

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

The dog was chasing a cat.

I

dog

joe

run-02

location

garden

semantic parsing

see-01

ARG0

ARG1

poss

chase-01

arg0

arg1

cat

dog

Why Graph-Based Meaning Representation?

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

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

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

I saw Joe’s dog, which was running in the garden.
The dog was chasing a cat.
Why Graph-Based Meaning Representation?

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

The dog was chasing a cat.

```
<table>
<thead>
<tr>
<th>semantic parsing</th>
</tr>
</thead>
<tbody>
<tr>
<td>see-01</td>
</tr>
<tr>
<td>ARG0</td>
</tr>
<tr>
<td>dog</td>
</tr>
<tr>
<td>ARG1</td>
</tr>
<tr>
<td>Joe</td>
</tr>
<tr>
<td>run-02</td>
</tr>
<tr>
<td>location</td>
</tr>
<tr>
<td>garden</td>
</tr>
<tr>
<td>chase-01</td>
</tr>
<tr>
<td>ARG0</td>
</tr>
<tr>
<td>dog</td>
</tr>
<tr>
<td>ARG1</td>
</tr>
<tr>
<td>cat</td>
</tr>
</tbody>
</table>

merge

| see-01           |
| ARG0             |
| dog              |
| ARG1             |
| Joe              |
| run-02           |
| location         |
| garden           |
| chase-01         |
| ARG0             |
| dog              |
| ARG1             |
| cat              |

summarize

| chase-01         |
| ARG0             |
| location         |
| ARG1             |
| dog              |
| garden           |
| cat              |
| Joe              |

Why Graph-Based Meaning Representation?

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

The dog was chasing a cat.

Joe’s dog was chasing a cat in the garden.
Why Graph-Based Meaning Representation?

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

The dog was chasing a cat.

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

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

\[
\exists x : \text{technique}'(x) \land \text{similar}'(x), \exists y : \text{crop}'(y) \land \text{other}'(y) \\
\rightarrow \text{almost}'(\neg \text{possible}'(\text{apply}'(\_, x, y)))
\]
A similar technique is almost impossible to apply to other crops.

\[ \exists x : \text{technique}'(x) \land \text{similar}'(x), \exists y : \text{crop}'(y) \land \text{other}'(y) \land \rightarrow \text{almost}'(\neg \text{possible}'(\text{apply}'(\_, x, y))) \]
A similar technique is almost impossible to apply to other crops.

\[ \exists x : \text{technique}'(x) \land \text{similar}'(x), \exists y : \text{crop}'(y) \land \text{other}'(y) \rightarrow \text{almost}'(\neg \text{possible}'(\text{apply}'(\_ , x , y))) \]
A similar technique is almost impossible to apply to other crops.

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

\[ \rightarrow \text{almost}'(\neg \text{possible}'(\text{apply}'(_, x, y))) \]
A similar technique is almost impossible to apply to other crops.

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

Different Desiderata and Levels of Abstraction

- Grammaticality (e.g. subject–verb agreement) vs. relational structure.
Semi-Formally: Trees vs. Graphs

Structural Wellformedness Conditions on Trees

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

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

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

Beyond Trees: General Graphs

- Argument sharing: nodes with multiple incoming edges (in-degree > 1);
Semi-Formally: Trees vs. Graphs

**Structural Wellformedness Conditions on Trees**
- Unique root, connected, single parent, free of cycles; maybe: projective;
- all nodes (but the root) reachable by unique directed path from root.

**Beyond Trees: General Graphs**
- **Argument sharing**: nodes with multiple incoming edges (*in*-degree $\geq 1$);
- some surface tokens do not contribute (as nodes; many function words);
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 $\geq 1$);
- some surface tokens do not contribute (as nodes; many function words);
- (structurally) multi-rooted: more than one node with zero $in$-degree;
Semi-Formally: Trees vs. Graphs

Structural Wellformedness Conditions on Trees

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

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

Beyond Trees: General Graphs

- Argument sharing: nodes with multiple incoming edges (\textit{in}-degree $> 1$);
- some surface tokens do not contribute (as nodes; many function words);
- (structurally) multi-rooted: more than one node with zero \textit{in}-degree;
- massive growth in modeling and algorithmic complexity (NP-complete).
## High-Level Goals of the Shared Task

### Cross-Framework Comparability and Interoperability

- Vast, **complex landscape** of representing natural language meaning;
- diverse linguistic traditions, modeling assumptions, levels of ambition;

→ clarify concepts and terminology; unify representations and evaluation.
High-Level Goals of the Shared Task

Cross-Framework Comparability and Interoperability
- Vast, complex landscape of representing natural language meaning;
- diverse linguistic traditions, modeling assumptions, levels of ambition;
→ clarify concepts and terminology; unify representations and evaluation.

Parsing into Graph-Structured Representations
- Cottage industry of parsers with output structures beyond rooted trees;
- distinct techniques, e.g. based on transitions, composition, ‘translation’;
- much framework-internal evolution: design reflects specific assumptions;
→ evaluate across frameworks; learning from complementary knowledge.
## High-Level Goals of the Shared Task

### Cross-Framework Comparability and Interoperability
- Vast, complex landscape of representing natural language meaning;
- diverse linguistic traditions, modeling assumptions, levels of ambition;
→ clarify concepts and terminology; unify representations and evaluation.

### Parsing into Graph-Structured Representations
- Cottage industry of parsers with output structures beyond rooted trees;
- distinct techniques, e.g. based on transitions, composition, ‘translation’;
- much framework-internal evolution: design reflects specific assumptions;
→ evaluate across frameworks; learning from complementary knowledge.

### Learning from Complementary Knowledge
- Cross-Framework Perspective: Seek commonality and complementarity.
Graph Theory 101

\[ G = \langle N, E, T \rangle \]

- \( G \) 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;
- semantic graphs often multi-rooted: rootness just a structural property;
- a node \( n \) is reentrant if \( \text{in}(n) > 1 \) (shared argument across predicates);
- 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;
- \( G \) is a tree if \( |T| = 1 \) and there is a unique path to all other nodes.
\( G = \langle N, E, T \rangle \)

- \( G \) 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);
- \( \text{in-} \) and \( \text{out-} \)-degree of \( n \in N \) count edges to and from \( n \); \( \text{in} = 0 \): root;
- top in \textit{Abrams arrived quickly.} is \textit{arrive}, but can be argument of \textit{quick};
- semantic graphs often \textit{multi-rooted}: rootness just a structural property;
Graph Theory 101

\[ G = \langle N, E, T \rangle \]

- \( G \) 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);
- \( \text{in} \) - and \( \text{out} \)-degree of \( n \in N \) count edges to and from \( n \); \( \text{in} = 0 \): 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 \( \text{in}(n) > 1 \) (shared argument across predicates);
Graph Theory 101

\[ G = \langle N, E, T \rangle \]

- **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 \( n \); \( \text{in} = 0 \): 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 \( \text{in}(n) > 1 \) (shared argument across predicates);
- **cycles** can occur: directed path from \( m \) to \( n \) and (‘back’) from \( n \) to \( m \);
Graph Theory 101

\[ G = \langle N, E, T \rangle \]

- \( G \) 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);
- \( \text{in} \)- and \( \text{out} \)-degree of \( n \in N \) count edges to and from \( n \); \( \text{in} = 0 \): 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 \( \text{in}(n) > 1 \) (shared argument across predicates);
- 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;
\[ G = \langle N, E, T \rangle \]

- \( G \) 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);
- \textit{in}- and \textit{out}-degree of \( n \in N \) count edges to and from \( n \); \( \text{in} = 0 \): root;
- top in \textit{Abrams arrived quickly}. is \textit{arrive}, but can be argument of \textit{quick};
- semantic graphs often multi-rooted: rootness just a structural property;
- a node \( n \) is reentrant if \( \text{in}(n) > 1 \) (shared argument across predicates);
- 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;
- \( G \) is a tree if \( |T| = 1 \) and there is a unique path to all other nodes.
Anchoring in the Surface String

Relating Pieces of Meaning to the Linguistic Signal

- Intuitively, sub-structures of meaning relate to sub-parts of the input;
- Semantic frameworks vary in how much weight to put on this relation;
Anchoring in the Surface String

Relating Pieces of Meaning to the Linguistic Signal

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

<table>
<thead>
<tr>
<th>Flavor Name</th>
<th>Example Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>(0)</td>
<td>bilexical DM, PSD nodes are sub-set of surface tokens</td>
</tr>
<tr>
<td>(1)</td>
<td>anchored EDS, UCCA free node–sub-string correspondences</td>
</tr>
<tr>
<td>(2)</td>
<td>unanchored AMR no explicit sub-string correspondences</td>
</tr>
</tbody>
</table>
Anchoring in the Surface String

Relating Pieces of Meaning to the Linguistic Signal

- Intuitively, sub-structures of meaning relate to sub-parts of the input;
- semantic frameworks vary in how much weight to put on this relation;
- **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;

<table>
<thead>
<tr>
<th>Flavor</th>
<th>Name</th>
<th>Example</th>
<th>Type of Anchoring</th>
</tr>
</thead>
<tbody>
<tr>
<td>(0)</td>
<td>bilexical</td>
<td>DM, PSD</td>
<td>nodes are sub-set of surface tokens</td>
</tr>
<tr>
<td>(1)</td>
<td>anchored</td>
<td>EDS, UCCA</td>
<td>free node–sub-string correspondences</td>
</tr>
<tr>
<td>(2)</td>
<td>unanchored</td>
<td>AMR</td>
<td>no explicit sub-string correspondences</td>
</tr>
</tbody>
</table>
Anchoring in the Surface String

Relating Pieces of Meaning to the Linguistic Signal

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

<table>
<thead>
<tr>
<th>Flavor</th>
<th>Name</th>
<th>Example</th>
<th>Type of Anchoring</th>
</tr>
</thead>
<tbody>
<tr>
<td>(0)</td>
<td>bilexical</td>
<td>DM, PSD</td>
<td>nodes are sub-set of surface tokens</td>
</tr>
<tr>
<td>(1)</td>
<td>anchored</td>
<td>EDS, UCCA</td>
<td>free node–sub-string correspondences</td>
</tr>
<tr>
<td>(2)</td>
<td>unanchored</td>
<td>AMR</td>
<td>no explicit sub-string correspondences</td>
</tr>
</tbody>
</table>
A Selection of Semantic 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;

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

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

<table>
<thead>
<tr>
<th>Selection Criteria</th>
</tr>
</thead>
<tbody>
<tr>
<td>‘Full-sentence’ semantics: all content-bearing units receive annotations;</td>
</tr>
<tr>
<td>natively graph-based: meaning representation through (directed) graphs;</td>
</tr>
<tr>
<td>large-scale, gold-standard annotations and parsers are publicly available;</td>
</tr>
<tr>
<td>five distinct frameworks, bi-lexical to unanchored; sadly, English only.</td>
</tr>
</tbody>
</table>

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

▶ PropBank (Palmer et al., 2005), Framenet (Baker et al., 1998), ...
▶ Groningen Parallel Meaning Bank: GMB, PMB (Basile et al., 2012)
▶ Universal Decompositional Semantics (White et al., 2016)
▶ Enhanced Universal Dependencies (Schuster & Manning, 2016)
▶ ...
A Selection of Semantic 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;
→ five distinct frameworks, bi-lexical to unanchored; sadly, English only.

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

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

...
A Selection of Semantic 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;

→ five distinct frameworks, bi-lexical to unanchored; sadly, English only.

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

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

- Two decades of great advances in syntactic dependencies and parsing;
- recently, renewed interest in meaning; algorithmic interest in graphs;
Arguably Basicest: Bi-lexical Semantic Dependencies

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

A similar technique is almost impossible to apply to other crops.
Arguably Basicest: Bi-lexical Semantic Dependencies

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

A similar technique is almost impossible to apply to other crops.
Arguably Basicest: Bi-lexical Semantic Dependencies

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

A similar technique is almost impossible to apply to other crops.
Arguably Basicest: Bi-lexical Semantic Dependencies

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

A similar technique is almost impossible to apply to other crops.
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, frames; edges encode argument roles and maybe some construction semantics;

A similar technique is almost impossible to apply to other crops.
Arguably Basicest: Bi-lexical Semantic Dependencies

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

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

A similar technique is almost impossible to apply to other crops.
Two Bi-Lexical Frameworks: DM & PSD

DM: DELPH-IN MRS Bi-Lexical Dependencies (Ivanova et al., 2012)

- Simplification from underspecified logical forms (ERS; coming later);

PSD: Prague Semantic Dependencies (Hajič et al., 2012)

- Simplification from FGD tectogrammatical trees (Sgall et al., 1986).

Diagram:

- BV
- ARG1
- ARG2
- ARG3
- top
- a
- similar
- technique
- almost
- impossible
- apply
- other
- crop
Two Bi-Lexical Frameworks: DM & PSD

**DM: DELPH-IN MRS Bi-Lexical Dependencies (Ivanova et al., 2012)**

- Simplification from underspecified logical forms (ERS; coming later);

**PSD: Prague Semantic Dependencies (Hajič et al., 2012)**

- Simplification from FGD tectogrammatical trees (Sgall et al., 1986).
A similar technique is almost impossible to apply to other crops.
A similar technique is almost impossible to apply to other crops.
Multi-Layered Design (Abend & Rappoport, 2013); **Foundational Layer**

▶ Tree backbone: semantic ‘constituents’ are **scenes** (‘clauses’) and **units**;

A similar technique is almost impossible to apply to other crops.
Multi-Layered Design (Abend & Rappoport, 2013); **Foundational Layer**

- Tree backbone: semantic ‘constituents’ are scenes (‘clauses’) and units;
- scenes (Process or State): pArticipants and aDverbials (plus F and U);

*A similar technique is almost impossible to apply to other crops.*
A similar technique is almost impossible to apply to other crops.
A similar technique is almost impossible to apply to other crops.
### Training and Evaluation Data in the Shared Task

<table>
<thead>
<tr>
<th>Flavor</th>
<th>DM</th>
<th>PSD</th>
<th>EDS</th>
<th>UCCA</th>
<th>AMR</th>
</tr>
</thead>
<tbody>
<tr>
<td>train</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>Text Type</td>
<td>newspaper</td>
<td>newspaper</td>
<td>newspaper</td>
<td>mixed</td>
<td>mixed</td>
</tr>
<tr>
<td>Sentences</td>
<td>35,656</td>
<td>35,656</td>
<td>35,656</td>
<td>6,572</td>
<td>56,240</td>
</tr>
<tr>
<td>Tokens</td>
<td>802,717</td>
<td>802,717</td>
<td>802,717</td>
<td>138,268</td>
<td>1,000,217</td>
</tr>
<tr>
<td>test</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Text Type</td>
<td>mixed</td>
<td>mixed</td>
<td>mixed</td>
<td>mixed</td>
<td>mixed</td>
</tr>
<tr>
<td>Sentences</td>
<td>3,359</td>
<td>3,359</td>
<td>3,359</td>
<td>1,131</td>
<td>1,998</td>
</tr>
<tr>
<td>Tokens</td>
<td>64,853</td>
<td>64,853</td>
<td>64,853</td>
<td>21,647</td>
<td>39,520</td>
</tr>
</tbody>
</table>

- DM, PSD, and ESD annotate the same text (Sections 00–20 of WSJ);
- UCCA: samples of EWT & Wikipedia; AMR: twelve different sources;
## Training and Evaluation Data in the Shared Task

<table>
<thead>
<tr>
<th></th>
<th>DM</th>
<th>PSD</th>
<th>EDS</th>
<th>UCCA</th>
<th>AMR</th>
</tr>
</thead>
<tbody>
<tr>
<td>Flavor</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>Text Type</td>
<td>newspaper</td>
<td>newspaper</td>
<td>newspaper</td>
<td>mixed</td>
<td>mixed</td>
</tr>
<tr>
<td>Sentences</td>
<td>35,656</td>
<td>35,656</td>
<td>35,656</td>
<td>6,572</td>
<td>56,240</td>
</tr>
<tr>
<td>Tokens</td>
<td>802,717</td>
<td>802,717</td>
<td>802,717</td>
<td>138,268</td>
<td>1,000,217</td>
</tr>
<tr>
<td>Text Type</td>
<td>mixed</td>
<td>mixed</td>
<td>mixed</td>
<td>mixed</td>
<td>mixed</td>
</tr>
<tr>
<td>Sentences</td>
<td>3,359</td>
<td>3,359</td>
<td>3,359</td>
<td>1,131</td>
<td>1,998</td>
</tr>
<tr>
<td>Tokens</td>
<td>64,853</td>
<td>64,853</td>
<td>64,853</td>
<td>21,647</td>
<td>39,520</td>
</tr>
</tbody>
</table>

- DM, PSD, and ESD annotate the same text (Sections 00–20 of WSJ);
- UCCA: samples of EWT & Wikipedia; AMR: twelve different sources;
- linguistics: 100-item WSJ sample in all frameworks publicly available;
## Training and Evaluation Data in the Shared Task

<table>
<thead>
<tr>
<th>Flavor</th>
<th>DM</th>
<th>PSD</th>
<th>EDS</th>
<th>UCCA</th>
<th>AMR</th>
</tr>
</thead>
<tbody>
<tr>
<td>train</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>Text Type</td>
<td>newspaper</td>
<td>newspaper</td>
<td>newspaper</td>
<td>mixed</td>
<td>mixed</td>
</tr>
<tr>
<td>Sentences</td>
<td>35,656</td>
<td>35,656</td>
<td>35,656</td>
<td>6,572</td>
<td>56,240</td>
</tr>
<tr>
<td>Tokens</td>
<td>802,717</td>
<td>802,717</td>
<td>802,717</td>
<td>138,268</td>
<td>1,000,217</td>
</tr>
<tr>
<td>test</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Text Type</td>
<td>mixed</td>
<td>mixed</td>
<td>mixed</td>
<td>mixed</td>
<td>mixed</td>
</tr>
<tr>
<td>Sentences</td>
<td>3,359</td>
<td>3,359</td>
<td>3,359</td>
<td>1,131</td>
<td>1,998</td>
</tr>
<tr>
<td>Tokens</td>
<td>64,853</td>
<td>64,853</td>
<td>64,853</td>
<td>21,647</td>
<td>39,520</td>
</tr>
</tbody>
</table>

- DM, PSD, and ESD annotate the same text (Sections 00–20 of WSJ);
- UCCA: samples of EWT & Wikipedia; AMR: twelve different sources;
- linguistics: 100-item WSJ sample in all frameworks publicly available;
- evaluation: subset of 100 sentences from *The Little Prince* is public.
## Graphbank Statistics (Kuhlmann & Oepen, 2016)

<table>
<thead>
<tr>
<th></th>
<th>DM</th>
<th>PSD</th>
<th>EDS</th>
<th>UCCA</th>
<th>AMR$^{-1}$</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Counts</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(01) number of graphs</td>
<td>35,656</td>
<td>35,656</td>
<td>35,656</td>
<td>6,572</td>
<td>56,240</td>
</tr>
<tr>
<td>(01) number of tokens</td>
<td>802,717</td>
<td>802,717</td>
<td>802,717</td>
<td>138,268</td>
<td>1,000,217</td>
</tr>
<tr>
<td>(02) average number of tokens</td>
<td>22.51</td>
<td>22.51</td>
<td>22.51</td>
<td>21.03</td>
<td>17.78</td>
</tr>
<tr>
<td>(03) average nodes per token</td>
<td>0.77</td>
<td>0.64</td>
<td>1.29</td>
<td>1.37</td>
<td>0.65</td>
</tr>
<tr>
<td>(04) number of edge labels</td>
<td>59</td>
<td>90</td>
<td>10</td>
<td>15</td>
<td>101</td>
</tr>
<tr>
<td>(05) $%_g$ trees</td>
<td>2.31</td>
<td>42.26</td>
<td>0.09</td>
<td>34.83</td>
<td>22.24</td>
</tr>
<tr>
<td>(06) $%_g$ treewidth one</td>
<td>69.82</td>
<td>43.08</td>
<td>68.99</td>
<td>41.57</td>
<td>50.00</td>
</tr>
<tr>
<td>(07) average treewidth</td>
<td>1.30</td>
<td>1.61</td>
<td>1.31</td>
<td>1.61</td>
<td>1.56</td>
</tr>
<tr>
<td>(08) maximal treewidth</td>
<td>3</td>
<td>7</td>
<td>3</td>
<td>4</td>
<td>5</td>
</tr>
<tr>
<td>(09) average edge density</td>
<td>1.019</td>
<td>1.073</td>
<td>1.015</td>
<td>1.053</td>
<td>1.092</td>
</tr>
<tr>
<td>(10) $%_n$ reentrant</td>
<td>27.43</td>
<td>11.41</td>
<td>32.78</td>
<td>4.98</td>
<td>19.89</td>
</tr>
<tr>
<td>(11) $%_g$ cyclic</td>
<td>0.00</td>
<td>0.00</td>
<td>0.12</td>
<td>0.00</td>
<td>0.38</td>
</tr>
<tr>
<td>(12) $%_g$ not connected</td>
<td>6.57</td>
<td>0.70</td>
<td>1.74</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>(13) $%_g$ multi-rooted</td>
<td>97.47</td>
<td>40.60</td>
<td>99.93</td>
<td>0.00</td>
<td>71.37</td>
</tr>
<tr>
<td>(14) percentage non-top roots</td>
<td>44.94</td>
<td>4.34</td>
<td>54.85</td>
<td>0.00</td>
<td>20.09</td>
</tr>
<tr>
<td><strong>Treeness</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(15) average edge length</td>
<td>2.684</td>
<td>3.320</td>
<td>–</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>(16) $%_g$ noncrossing</td>
<td>69.21</td>
<td>64.61</td>
<td>–</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>(17) $%_g$ pagonenumber two</td>
<td>99.59</td>
<td>98.08</td>
<td>–</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>Counts</td>
<td>DM</td>
<td>PSD</td>
<td>EDS</td>
<td>UCCA</td>
<td>AMR$^{-1}$</td>
</tr>
<tr>
<td>-----------------</td>
<td>--------</td>
<td>--------</td>
<td>--------</td>
<td>--------</td>
<td>------------</td>
</tr>
<tr>
<td>(01) number of graphs</td>
<td>35,656</td>
<td>35,656</td>
<td>35,656</td>
<td>6,572</td>
<td>56,240</td>
</tr>
<tr>
<td>(01) number of tokens</td>
<td>802,717</td>
<td>802,717</td>
<td>802,717</td>
<td>138,268</td>
<td>1,000,217</td>
</tr>
<tr>
<td>(02) average number of tokens</td>
<td>22.51</td>
<td>22.51</td>
<td>22.51</td>
<td>21.03</td>
<td>17.78</td>
</tr>
<tr>
<td>(03) average nodes per token</td>
<td>0.77</td>
<td>0.64</td>
<td>1.29</td>
<td>1.37</td>
<td>0.65</td>
</tr>
<tr>
<td>(04) number of edge labels</td>
<td>59</td>
<td>90</td>
<td>10</td>
<td>15</td>
<td>101</td>
</tr>
<tr>
<td>(05) %$_g$ trees</td>
<td>2.31</td>
<td>42.26</td>
<td>0.09</td>
<td>34.83</td>
<td>22.24</td>
</tr>
<tr>
<td>(06) %$_g$ treewidth one</td>
<td>69.82</td>
<td>43.08</td>
<td>68.99</td>
<td>41.57</td>
<td>50.00</td>
</tr>
<tr>
<td>(07) average treewidth</td>
<td>1.30</td>
<td>1.61</td>
<td>1.31</td>
<td>1.61</td>
<td>1.56</td>
</tr>
<tr>
<td>(08) maximal treewidth</td>
<td>3</td>
<td>7</td>
<td>3</td>
<td>4</td>
<td>5</td>
</tr>
<tr>
<td>(09) average edge density</td>
<td>1.019</td>
<td>1.073</td>
<td>1.015</td>
<td>1.053</td>
<td>1.092</td>
</tr>
<tr>
<td>(10) %$_r$ reentrant</td>
<td>27.43</td>
<td>11.41</td>
<td>32.78</td>
<td>4.98</td>
<td>19.89</td>
</tr>
<tr>
<td>(11) %$_g$ cyclic</td>
<td>0.00</td>
<td>0.00</td>
<td>0.12</td>
<td>0.00</td>
<td>0.38</td>
</tr>
<tr>
<td>(12) %$_g$ not connected</td>
<td>6.57</td>
<td>0.70</td>
<td>1.74</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>(13) %$_g$ multi-rooted</td>
<td>97.47</td>
<td>40.60</td>
<td>99.93</td>
<td>0.00</td>
<td>71.37</td>
</tr>
<tr>
<td>(14) percentage non-top roots</td>
<td>44.94</td>
<td>4.34</td>
<td>54.85</td>
<td>0.00</td>
<td>20.09</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Treeness</th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>(15) average edge length</td>
<td>2.684</td>
<td>3.320</td>
<td>–</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>(16) %$_g$ noncrossing</td>
<td>69.21</td>
<td>64.61</td>
<td>–</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>(17) %$_g$ pageneumber two</td>
<td>99.59</td>
<td>98.08</td>
<td>–</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>Counts</td>
<td>DM</td>
<td>PSD</td>
<td>EDS</td>
<td>UCCA</td>
<td>AMR(^{-1})</td>
</tr>
<tr>
<td>------------------------------------</td>
<td>---------</td>
<td>---------</td>
<td>---------</td>
<td>---------</td>
<td>--------------</td>
</tr>
<tr>
<td>(01) number of graphs</td>
<td>35,656</td>
<td>35,656</td>
<td>35,656</td>
<td>6,572</td>
<td>56,240</td>
</tr>
<tr>
<td>(01) number of tokens</td>
<td>802,717</td>
<td>802,717</td>
<td>802,717</td>
<td>138,268</td>
<td>1,000,217</td>
</tr>
<tr>
<td>(02) average number of tokens</td>
<td>22.51</td>
<td>22.51</td>
<td>22.51</td>
<td>21.03</td>
<td>17.78</td>
</tr>
<tr>
<td>(03) average nodes per token</td>
<td>0.77</td>
<td>0.64</td>
<td>1.29</td>
<td>1.37</td>
<td>0.65</td>
</tr>
<tr>
<td>(04) number of edge labels</td>
<td>59</td>
<td>90</td>
<td>10</td>
<td>15</td>
<td>101</td>
</tr>
<tr>
<td>(05) (%)_g\ trees</td>
<td>2.31</td>
<td>42.26</td>
<td>0.09</td>
<td>34.83</td>
<td>22.24</td>
</tr>
<tr>
<td>(06) (%)_g\ treewidth one</td>
<td>69.82</td>
<td>43.08</td>
<td>68.99</td>
<td>41.57</td>
<td>50.00</td>
</tr>
<tr>
<td>(07) average treewidth</td>
<td>1.30</td>
<td>1.61</td>
<td>1.31</td>
<td>1.61</td>
<td>1.56</td>
</tr>
<tr>
<td>(08) maximal treewidth</td>
<td>3</td>
<td>7</td>
<td>3</td>
<td>4</td>
<td>5</td>
</tr>
<tr>
<td>(09) average edge density</td>
<td>1.019</td>
<td>1.073</td>
<td>1.015</td>
<td>1.053</td>
<td>1.092</td>
</tr>
<tr>
<td>(10) (%)_r\ reentrant</td>
<td>27.43</td>
<td>11.41</td>
<td>32.78</td>
<td>4.98</td>
<td>19.89</td>
</tr>
<tr>
<td>(11) (%)_g\ cyclic</td>
<td>0.00</td>
<td>0.00</td>
<td>0.12</td>
<td>0.00</td>
<td>0.38</td>
</tr>
<tr>
<td>(12) (%)_g\ not connected</td>
<td>6.57</td>
<td>0.70</td>
<td>1.74</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>(13) (%)_g\ multi-rooted</td>
<td>97.47</td>
<td>40.60</td>
<td>99.93</td>
<td>0.00</td>
<td>71.37</td>
</tr>
<tr>
<td>(14) percentage non-top roots</td>
<td>44.94</td>
<td>4.34</td>
<td>54.85</td>
<td>0.00</td>
<td>20.09</td>
</tr>
<tr>
<td>Treeness</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(15) average edge length</td>
<td>2.684</td>
<td>3.320</td>
<td>–</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>(16) (%)_g\ noncrossing</td>
<td>69.21</td>
<td>64.61</td>
<td>–</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>(17) (%)_g\ pagenumerator two</td>
<td>99.59</td>
<td>98.08</td>
<td>–</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>Counts</td>
<td>DM</td>
<td>PSD</td>
<td>EDS</td>
<td>UCCA</td>
<td>AMR(^{-1})</td>
</tr>
<tr>
<td>--------</td>
<td>----</td>
<td>-----</td>
<td>-----</td>
<td>------</td>
<td>-------------</td>
</tr>
<tr>
<td>(01)</td>
<td>number of graphs</td>
<td>35,656</td>
<td>35,656</td>
<td>35,656</td>
<td>6,572</td>
</tr>
<tr>
<td>(01)</td>
<td>number of tokens</td>
<td>802,717</td>
<td>802,717</td>
<td>802,717</td>
<td>138,268</td>
</tr>
<tr>
<td>(02)</td>
<td>average number of tokens</td>
<td>22.51</td>
<td>22.51</td>
<td>22.51</td>
<td>21.03</td>
</tr>
<tr>
<td>(03)</td>
<td>average nodes per token</td>
<td>0.77</td>
<td>0.64</td>
<td>1.29</td>
<td>1.37</td>
</tr>
<tr>
<td>(04)</td>
<td>number of edge labels</td>
<td>59</td>
<td>90</td>
<td>10</td>
<td>15</td>
</tr>
<tr>
<td>(05)</td>
<td>% trees</td>
<td>2.31</td>
<td>42.26</td>
<td>0.09</td>
<td>34.83</td>
</tr>
<tr>
<td>(06)</td>
<td>% treewidth one</td>
<td>69.82</td>
<td>43.08</td>
<td>68.99</td>
<td>41.57</td>
</tr>
<tr>
<td>(07)</td>
<td>average treewidth</td>
<td>1.30</td>
<td>1.61</td>
<td>1.31</td>
<td>1.61</td>
</tr>
<tr>
<td>(08)</td>
<td>maximal treewidth</td>
<td>3</td>
<td>7</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>(09)</td>
<td>average edge density</td>
<td>1.019</td>
<td>1.073</td>
<td>1.015</td>
<td>1.053</td>
</tr>
<tr>
<td>(10)</td>
<td>% reentrant</td>
<td>27.43</td>
<td>11.41</td>
<td>32.78</td>
<td>4.98</td>
</tr>
<tr>
<td>(11)</td>
<td>% cyclic</td>
<td>0.00</td>
<td>0.00</td>
<td>0.12</td>
<td>0.00</td>
</tr>
<tr>
<td>(12)</td>
<td>% not connected</td>
<td>6.57</td>
<td>0.70</td>
<td>1.74</td>
<td>0.00</td>
</tr>
<tr>
<td>(13)</td>
<td>% multi-rooted</td>
<td>97.47</td>
<td>40.60</td>
<td>99.93</td>
<td>0.00</td>
</tr>
<tr>
<td>(14)</td>
<td>percentage non-top roots</td>
<td>44.94</td>
<td>4.34</td>
<td>54.85</td>
<td>0.00</td>
</tr>
<tr>
<td>Order</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(15)</td>
<td>average edge length</td>
<td>2.684</td>
<td>3.320</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>(16)</td>
<td>% noncrossing</td>
<td>69.21</td>
<td>64.61</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>(17)</td>
<td>% pagonenumber two</td>
<td>99.59</td>
<td>98.08</td>
<td>–</td>
<td>–</td>
</tr>
</tbody>
</table>
Graphbank Statistics (Kuhlmann & Oepen, 2016)

<table>
<thead>
<tr>
<th>Counts</th>
<th>DM</th>
<th>PSD</th>
<th>EDS</th>
<th>UCCA</th>
<th>AMR⁻¹</th>
</tr>
</thead>
<tbody>
<tr>
<td>(01) number of graphs</td>
<td>35,656</td>
<td>35,656</td>
<td>35,656</td>
<td>6,572</td>
<td>56,240</td>
</tr>
<tr>
<td>(01) number of tokens</td>
<td>802,717</td>
<td>802,717</td>
<td>802,717</td>
<td>138,268</td>
<td>1,000,217</td>
</tr>
<tr>
<td>(02) average number of tokens</td>
<td>22.51</td>
<td>22.51</td>
<td>22.51</td>
<td>21.03</td>
<td>17.78</td>
</tr>
<tr>
<td>(03) average nodes per token</td>
<td>0.77</td>
<td>0.64</td>
<td>1.29</td>
<td>1.37</td>
<td>0.65</td>
</tr>
<tr>
<td>(04) number of edge labels</td>
<td>59</td>
<td>90</td>
<td>10</td>
<td>15</td>
<td>101</td>
</tr>
<tr>
<td>(05) %g trees</td>
<td>2.31</td>
<td>42.26</td>
<td>0.09</td>
<td>34.83</td>
<td>22.24</td>
</tr>
<tr>
<td>(06) %g treewidth one</td>
<td>69.82</td>
<td>43.08</td>
<td>68.99</td>
<td>41.57</td>
<td>50.00</td>
</tr>
<tr>
<td>(07) average treewidth</td>
<td>1.30</td>
<td>1.61</td>
<td>1.31</td>
<td>1.61</td>
<td>1.56</td>
</tr>
<tr>
<td>(08) maximal treewidth</td>
<td>3</td>
<td>7</td>
<td>3</td>
<td>4</td>
<td>5</td>
</tr>
<tr>
<td>(09) average edge density</td>
<td>1.019</td>
<td>1.073</td>
<td>1.015</td>
<td>1.053</td>
<td>1.092</td>
</tr>
<tr>
<td>(10) %n reentrant</td>
<td>27.43</td>
<td>11.41</td>
<td>32.78</td>
<td>4.98</td>
<td>19.89</td>
</tr>
<tr>
<td>(11) %g cyclic</td>
<td>0.00</td>
<td>0.00</td>
<td>0.12</td>
<td>0.00</td>
<td>0.38</td>
</tr>
<tr>
<td>(12) %g not connected</td>
<td>6.57</td>
<td>0.70</td>
<td>1.74</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>(13) %g multi-rooted</td>
<td>97.47</td>
<td>40.60</td>
<td>99.93</td>
<td>0.00</td>
<td>71.37</td>
</tr>
<tr>
<td>(14) percentage non-top roots</td>
<td>44.94</td>
<td>4.34</td>
<td>54.85</td>
<td>0.00</td>
<td>20.09</td>
</tr>
<tr>
<td>Order</td>
<td>DM</td>
<td>PSD</td>
<td>EDS</td>
<td>UCCA</td>
<td>AMR⁻¹</td>
</tr>
<tr>
<td>(15) average edge length</td>
<td>2.684</td>
<td>3.320</td>
<td>–</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>(16) %g noncrossing</td>
<td>69.21</td>
<td>64.61</td>
<td>–</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>(17) %g pagenumerator two</td>
<td>99.59</td>
<td>98.08</td>
<td>–</td>
<td>–</td>
<td>–</td>
</tr>
</tbody>
</table>
Cross-Framework Evaluation: MRP Graph Similarity

- Break down graphs into types of information: per-type and overall $F_1$;

Different Types of Semantic Graph ‘Atoms’

<table>
<thead>
<tr>
<th></th>
<th>DM</th>
<th>PSD</th>
<th>EDS</th>
<th>UCCA</th>
<th>AMR</th>
</tr>
</thead>
<tbody>
<tr>
<td>Top Nodes</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Labeled Edges</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Node Labels</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>X</td>
<td>✓</td>
</tr>
<tr>
<td>Node Properties</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>X</td>
<td>✓</td>
</tr>
<tr>
<td>Node Anchoring</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>X</td>
</tr>
<tr>
<td>Edge Attributes</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>✓</td>
<td>X</td>
</tr>
</tbody>
</table>
Cross-Framework Evaluation: MRP Graph Similarity

- Break down graphs into types of information: per-type and overall F1;
- tops

_Pierre retired._

_Different Types of Semantic Graph ‘Atoms’_

<table>
<thead>
<tr>
<th></th>
<th>DM</th>
<th>PSD</th>
<th>EDS</th>
<th>UCCA</th>
<th>AMR</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Top Nodes</strong></td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Labeled Edges</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Node Labels</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>X</td>
<td>✓</td>
</tr>
<tr>
<td>Node Properties</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>X</td>
<td>✓</td>
</tr>
<tr>
<td>Node Anchoring</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>X</td>
</tr>
<tr>
<td>Edge Attributes</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>✓</td>
<td>X</td>
</tr>
</tbody>
</table>
Cross-Framework Evaluation: MRP Graph Similarity

- Break down graphs into types of information: per-type and overall $F_1$;
- tops and (labeled) edges;

Different Types of Semantic Graph ‘Atoms’

<table>
<thead>
<tr>
<th></th>
<th>DM</th>
<th>PSD</th>
<th>EDS</th>
<th>UCCA</th>
<th>AMR</th>
</tr>
</thead>
<tbody>
<tr>
<td>Top Nodes</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Labeled Edges</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Node Labels</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>x</td>
<td>✓</td>
</tr>
<tr>
<td>Node Properties</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>x</td>
<td>✓</td>
</tr>
<tr>
<td>Node Anchoring</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>x</td>
</tr>
<tr>
<td>Edge Attributes</td>
<td>✓</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>✓</td>
</tr>
</tbody>
</table>

Pierre retired.
Cross-Framework Evaluation: MRP Graph Similarity

- Break down graphs into types of information: per-type and overall $F_1$;
- tops and (labeled) edges; labels,

_Pierre retired._
Cross-Framework Evaluation: MRP Graph Similarity

- Break down graphs into types of information: per-type and overall $F_1$;
- tops and (labeled) edges; labels, properties,

Pierre retired.
Cross-Framework Evaluation: MRP Graph Similarity

- Break down graphs into types of information: per-type and overall $F_1$;
- tops and (labeled) edges; labels, properties, anchors,

Different Types of Semantic Graph ‘Atoms’

<table>
<thead>
<tr>
<th></th>
<th>DM</th>
<th>PSD</th>
<th>EDS</th>
<th>UCCA</th>
<th>AMR</th>
</tr>
</thead>
<tbody>
<tr>
<td>Top Nodes</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Labeled Edges</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Node Labels</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>x</td>
<td>✓</td>
</tr>
<tr>
<td>Node Properties</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>x</td>
<td>✓</td>
</tr>
<tr>
<td><strong>Node Anchoring</strong></td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>x</td>
</tr>
<tr>
<td>Edge Attributes</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>✓</td>
<td>x</td>
</tr>
</tbody>
</table>

_Pierre retired._
Cross-Framework Evaluation: MRP Graph Similarity

- Break down graphs into types of information: per-type and overall $F_1$;
- tops and (labeled) edges; labels, properties, anchors, and attributes;

Pierre retired.

Different Types of Semantic Graph ‘Atoms’

<table>
<thead>
<tr>
<th></th>
<th>DM</th>
<th>PSD</th>
<th>EDS</th>
<th>UCCA</th>
<th>AMR</th>
</tr>
</thead>
<tbody>
<tr>
<td>Top Nodes</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Labeled Edges</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Node Labels</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>X</td>
<td>✓</td>
</tr>
<tr>
<td>Node Properties</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>X</td>
<td>✓</td>
</tr>
<tr>
<td>Node Anchoring</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>X</td>
</tr>
<tr>
<td>Edge Attributes</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>✓</td>
<td>X</td>
</tr>
</tbody>
</table>
Cross-Framework Evaluation: MRP Graph Similarity

- Break down graphs into types of information: per-type and overall $F_1$;
- tops and (labeled) edges; labels, properties, anchors, and attributes;
- requires node–node correspondences; search for overall maximum score;
- maximum common edge subgraph isomorphism (MCES) is NP-hard;

Different Types of Semantic Graph ‘Atoms’

<table>
<thead>
<tr>
<th></th>
<th>DM</th>
<th>PSD</th>
<th>EDS</th>
<th>UCCA</th>
<th>AMR</th>
</tr>
</thead>
<tbody>
<tr>
<td>Top Nodes</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Labeled Edges</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Node Labels</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>X</td>
<td>✓</td>
</tr>
<tr>
<td>Node Properties</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>X</td>
<td>✓</td>
</tr>
<tr>
<td>Node Anchoring</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>X</td>
</tr>
<tr>
<td>Edge Attributes</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>✓</td>
<td>X</td>
</tr>
</tbody>
</table>

Pierre retired.
Cross-Framework Evaluation: MRP Graph Similarity

- Break down graphs into types of information: per-type and overall $F_1$;
- tops and (labeled) edges; labels, properties, anchors, and attributes;
- requires node–node correspondences; search for overall maximum score;
- maximum common edge subgraph isomorphism (MCES) is NP-hard;
  → smart initialization, scheduling, and pruning yield strong approximation.

Different Types of Semantic Graph ‘Atoms’

<table>
<thead>
<tr>
<th></th>
<th>DM</th>
<th>PSD</th>
<th>EDS</th>
<th>UCCA</th>
<th>AMR</th>
</tr>
</thead>
<tbody>
<tr>
<td>Top Nodes</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Labeled Edges</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Node Labels</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>X</td>
<td>✓</td>
</tr>
<tr>
<td>Node Properties</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>X</td>
<td>✓</td>
</tr>
<tr>
<td>Node Anchoring</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>X</td>
</tr>
<tr>
<td>Edge Attributes</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>✓</td>
<td>X</td>
</tr>
</tbody>
</table>

_Pierre retired._
# High-Level Overview of Submissions

<table>
<thead>
<tr>
<th>Teams</th>
<th>DM</th>
<th>PSD</th>
<th>EDS</th>
<th>UCCA</th>
<th>AMR</th>
<th>MTL</th>
<th>Approach</th>
</tr>
</thead>
<tbody>
<tr>
<td>ERG</td>
<td>✓</td>
<td>X</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>X</td>
</tr>
<tr>
<td>TUPA</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓/✓</td>
<td>Composition</td>
</tr>
<tr>
<td>HIT-SCIR</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>X</td>
</tr>
<tr>
<td>SJTU–NICT</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>X</td>
<td>Transition</td>
</tr>
<tr>
<td>SUDA–Alibaba</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>Factorization</td>
</tr>
<tr>
<td>Saarland</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>X</td>
<td>Composition</td>
</tr>
<tr>
<td>Hitachi</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>Factorization</td>
</tr>
<tr>
<td>ÚFAL MRPipe</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>X</td>
</tr>
<tr>
<td>ShanghaiTech</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>X</td>
<td>Factorization</td>
</tr>
<tr>
<td>Amazon</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>X</td>
<td>✓</td>
<td>X</td>
<td>Factorization</td>
</tr>
<tr>
<td>JBNJU</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>X</td>
<td>X</td>
<td>Factorization</td>
</tr>
<tr>
<td>SJTU</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>Transition</td>
</tr>
<tr>
<td>ÚFAL–Oslo</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>X</td>
<td>Transition</td>
</tr>
<tr>
<td>HKUST</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>X</td>
<td>?</td>
<td>?</td>
</tr>
<tr>
<td>Bocharov</td>
<td>X</td>
<td>✓</td>
<td>✓</td>
<td>X</td>
<td>X</td>
<td>✓</td>
<td>?</td>
</tr>
<tr>
<td>Peking</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>CUHK</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>Transition</td>
</tr>
<tr>
<td>Anonymous</td>
<td>X</td>
<td>✓</td>
<td>✓</td>
<td>X</td>
<td>X</td>
<td>?</td>
<td>?</td>
</tr>
</tbody>
</table>
Score Distributions

Overall
DM
PSD
EDS
UCCA
AMR

Composition
Factorization
Transition
?
Composition-Based Approaches

- Explicitly modeling the derivation process.
- A parser evaluates a derivation licensed by a symbolic system.
Factorization-Based Approaches

- Inspired by graph-based dependency parsers.
- Explicitly modeling the target structure.
- A parser evaluates factors of a candidate graph.
Transition-Based Approaches

- Inspired by transition-based dependency parsers.
- Incremental (left-to-right, word-by-word).
- Partial parse constrains subsequent actions.
- Greedy/beam search to get a parse.
Score Distributions: Zoom In

- Overall
- DM
- PSD
- EDS
- UCCA
- AMR

Composition
Factorization
Transition
?
Official Leaderboard: All Evaluation Data

Full Evaluation MRP F-score
The Little Prince Subset MRP F-score
State of the Art

Submissions from established top-performing teams:

- ShanghaiTech (DM, PSD)
- Peking (EDS)
- SUDA–Alibaba (UCCA)
- Saarland (AMR)

Outperformed in most cases!
Limiting Factors in Comparison to State of the Art

- New cross-framework metric: MRP
Limiting Factors in Comparison to State of the Art

- New cross-framework metric: MRP
- Different task definition (DM, PSD: nodes, not just edges)
- Different evaluation set (EDS: not just WSJ)
- Different normalization (AMR: inverted edges)
- Revised and extended annotation (UCCA, AMR)
Limiting Factors in Comparison to State of the Art

- New cross-framework metric: MRP
- Different task definition (DM, PSD: nodes, not just edges)
- Different evaluation set (EDS: not just WSJ)
- Different normalization (AMR: inverted edges)
- Revised and extended annotation (UCCA, AMR)
- No gold tokenization (or tags or lemmas)!
Lessons Learned

- Great **community interest**: 160 subscribers; 38 data licenses (via LDC);
- **task complexity** is **technical barrier to entry**: 16 + 2 teams submitted;
### Interim Conclusions & Outlook

#### Lessons Learned

- **Great community interest**: 160 subscribers; 38 data licenses (via LDC);
- **task complexity is technical barrier to entry**: 16 + 2 teams submitted;
  - **advanced state of the art** on four frameworks (but possibly not AMR);

---

<table>
<thead>
<tr>
<th>Lessons Learned</th>
</tr>
</thead>
<tbody>
<tr>
<td>Great community interest: 160 subscribers; 38 data licenses (via LDC);</td>
</tr>
<tr>
<td>task complexity is technical barrier to entry: 16 + 2 teams submitted;</td>
</tr>
<tr>
<td>advanced state of the art on four frameworks (but possibly not AMR);</td>
</tr>
</tbody>
</table>
Lessons Learned

- **Great community interest**: 160 subscribers; 38 data licenses (via LDC);
- **task complexity is technical barrier to entry**: 16 + 2 teams submitted;
- advanced state of the art on four frameworks (but possibly not AMR);
- greatly increased cross-framework uniformity; but limited MTL so far.
Interim Conclusions & Outlook

Lessons Learned

▶ Great community interest: 160 subscribers; 38 data licenses (via LDC);
▶ task complexity is technical barrier to entry: 16 + 2 teams submitted;
→ advanced state of the art on four frameworks (but possibly not AMR);
→ greatly increased cross-framework uniformity; but limited MTL so far.

Outlook: Toward MRP 2020

▶ Invitation from SIGNLL to re-run (a follow-up variant) at CoNLL 2020;
Lessons Learned

► Great community interest: 160 subscribers; 38 data licenses (via LDC);

► task complexity is technical barrier to entry: 16 + 2 teams submitted;

→ advanced state of the art on four frameworks (but possibly not AMR);

→ greatly increased cross-framework uniformity; but limited MTL so far.

Outlook: Toward MRP 2020

► Invitation from SIGNLL to re-run (a follow-up variant) at CoNLL 2020;

? add Discourse Representation Graphs; maybe a few other languages;

? increased focus on evaluation metrics: score ‘larger pieces’; SEMBLEU;
Interim Conclusions & Outlook

Lessons Learned

▶ Great **community interest**: 160 subscribers; 38 data licenses (via LDC);
▶ task complexity is **technical barrier to entry**: 16 + 2 teams submitted;
→ advanced state of the art on four frameworks (but possibly not AMR);
→ greatly increased **cross-framework uniformity**; but **limited MTL** so far.

Outlook: Toward MRP 2020

▶ Invitation from SIGNLL to **re-run** (a follow-up variant) at CoNLL 2020;
? add Discourse Representation Graphs; maybe a few **other languages**;
? increased focus on **evaluation metrics**: score ‘larger pieces’; SEMBLEU;
→ **open discussion** with 2019 participants towards the end of this session.
Acknowledgments

Mohit Bansal, Emily M. Bender, Xavier Carreras, Jayeol Chun, Dotan Dvir, Dan Flickinger, Julia Hockenmaier, Andrey Kutuzov, Sebastian Schuster, Milan Straka, Zdeňka Urešová, and Aline Villavicencio

Linguistic Data Consortium,
Nordic Language Processing Laboratory
References I


TUPA at MRP 2019
A Multi-Task Baseline System

Daniel Hershcovich\textsuperscript{1}, Ofir Arviv\textsuperscript{2}

\textsuperscript{1}University of Copenhagen
\textsuperscript{2}Hebrew University of Jerusalem

CoNLL Shared Task
3 November 2019
After the break, you should look at my poster.

(Hershcovich et al., 2017)
Scarcity of Training Data

UCCA
AMR
DM
Multi-task improves UCCA parsing (Hershcovich et al., 2018).
After the break, you should look at my poster.

Multi-task improves UCCA parsing (Hershcovich et al., 2018).
After graduation, John moved to New York City.

Intermediate graph representation, extended transition system.
Transition Classifier

**BiLSTM + BERT (Devlin et al., 2019).**
BiLSTM + BERT (Devlin et al., 2019).
Results

Baseline: single-task + multi-task.

Full Evaluation MRP F-score (%). TUPA scores are post-evaluation.
Results

Baseline: single-task + multi-task.

Full Evaluation MRP F-score (%). TUPA scores are post-evaluation.
TUPA at MRP 2019
A Multi-Task Baseline System

Daniel Hershcovitch\textsuperscript{1}, Ofir Arviv\textsuperscript{2}

\textsuperscript{1}University of Copenhagen
\textsuperscript{2}Hebrew University of Jerusalem

CoNLL Shared Task
3 November 2019
References


The ERG at MRP 2019
Radically Compositional
Semantic Dependencies

http://erg.delph-in.net

Stephan Oepen♣ and Dan Flickinger♠
♣ University of Oslo, Department of Informatics
♠ Stanford University, Center for the Study of Language and Information

oe@ifi.uio.no, danf@stanford.edu
Background: English Resource Semantics (ERS)

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

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

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

- Grammar-based annotation: select 'correct' reading from parse forest;
- version 1214: some 85,000 annotated sentences, six different domains;
- Bender et al. (2015) report inter-annotator agreement of 0.94 EDM;
- EDS: graph-based simplification of ERS; DM: its bi-lexical 'reduction';

PET Unification-Based Parser (Callmeier, 2002)

- Highly optimized chart parser; (exact) $n$-best MaxEnt parse selection.
**LinGO English Resource Grammar (Flickinger, 2000; Flickinger et al., 2017)**

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

<table>
<thead>
<tr>
<th><strong>LinGO English Resource Grammar (Flickinger, 2000; Flickinger et al., 2017)</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>▶ Hand-designed <strong>computational grammar</strong> for English in HPSG framework;</td>
</tr>
<tr>
<td>▶ declarative, unification-based: parsing and realization; multiple engines;</td>
</tr>
<tr>
<td>▶ 25+ person years; coverage of 85–95% of running text across domains;</td>
</tr>
<tr>
<td>▶ underspecified meaning representation in MRS (Copestake et al., 2005).</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>LinGO Redwoods Treebank (Carter, 1997; Oepen et al., 2004)</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>▶ Grammar-based annotation: select ‘correct’ reading from parse forest;</td>
</tr>
</tbody>
</table>
LinGO English Resource Grammar (Flickinger, 2000; Flickinger et al., 2017)
▶ Hand-designed computational grammar for English in HPSG framework;
▶ declarative, unification-based: parsing and realization; multiple engines;
▶ 25+ person years; coverage of 85–95% of running text across domains;
▶ underspecified meaning representation in MRS (Copestake et al., 2005).

LinGO Redwoods Treebank (Carter, 1997; Oepen et al., 2004)
▶ Grammar-based annotation: select ‘correct’ reading from parse forest;
▶ version 1214: some 85,000 annotated sentences, six+ different domains;
Background: English Resource Semantics (ERS)

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

**LinGO Redwoods Treebank (Carter, 1997; Oepen et al., 2004)**
- Grammar-based annotation: select ‘correct’ reading from parse forest;
- version 1214: some 85,000 annotated sentences, six+ different domains;
- Bender et al. (2015) report **inter-annotator agreement** of 0.94 EDM$_{na}$;
## Background: English Resource Semantics (ERS)

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

### LinGO Redwoods Treebank (Carter, 1997; Oepen et al., 2004)
- Grammar-based annotation: select ‘correct’ reading from parse forest;
- version 1214: some 85,000 annotated sentences, six$^+$ different domains;
- Bender et al. (2015) report inter-annotator agreement of 0.94 EDM$_{na}$;
- EDS: graph-based **simplification** of ERS; DM: its bi-lexical ‘reduction’;
### Background: English Resource Semantics (ERS)

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

#### LinGO Redwoods Treebank (Carter, 1997; Oepen et al., 2004)
- Grammar-based annotation: select ‘correct’ reading from parse forest;
- version 1214: some 85,000 annotated sentences, six+ different domains;
- Bender et al. (2015) report inter-annotator agreement of 0.94 EDM_{na};
- EDS: graph-based simplification of ERS; DM: its bi-lexical ‘reduction’;

#### PET Unification-Based Parser (Callmeier, 2002)
- Highly optimized **chart parser**; (exact) n-best MaxEnt parse selection.
EDSs are ‘Radically Compositional’

Named CARG Pierre

Named CARG Vinken

compound

ARG2

ARG1

Pierre Vinken
EDSs are ‘Radically Compositional’

Named Entities

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

Michelle and Barack Obama

Pierre Vinken
EDSs are ‘Radically Compositional’

Named Entities

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

Michelle and Barack Obama

Pierre Vinken

on Monday
EDSs are ‘Radically Compositional’

**Named Entities**
- Underspecified structure in names;
- Few, lexically determined sub-types.

*Michelle and Barack Obama*

**Prepositions (and Similar)**
- Predicates: distinct two-place relation;
- Specialized sub-senses as appropriate.

*before and during the meeting*
EDSs are ‘Radically Compositional’

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

Michelle and Barack Obama

**Prepositions (and Similar)**
- Predicates: distinct two-place relation;
- specialized sub-senses as appropriate.

before and during the meeting

thirty-two
EDSs are ‘Radically Compositional’

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

*Michelle and Barack Obama*

**Prepositions (and Similar)**
- Predicates: distinct two-place relation;
- specialized sub-senses as appropriate.

*before and during the meeting*

**Literal Numbers**
- syntax yields arithmetic expressions;
- trivial ‘downstream’ normalization.

*ten to twenty thousand*
## Comparison to Top-Performing Data-Driven Parsers

<table>
<thead>
<tr>
<th></th>
<th>Tops</th>
<th></th>
<th></th>
<th>Labels</th>
<th></th>
<th></th>
<th>Properties</th>
<th></th>
<th>Anchors</th>
<th></th>
<th>Edges</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>P</td>
<td>R</td>
<td>F₁</td>
<td>P</td>
<td>R</td>
<td>F₁</td>
<td>P</td>
<td>R</td>
<td>F₁</td>
<td>P</td>
<td>R</td>
<td>F₁</td>
</tr>
<tr>
<td>DM</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ERG</td>
<td>.92</td>
<td>.92</td>
<td>.918</td>
<td>.99</td>
<td>.99</td>
<td>.987</td>
<td>.96</td>
<td>.96</td>
<td>.956</td>
<td>.99</td>
<td>.99</td>
<td>.994</td>
</tr>
<tr>
<td>SJTU–NICT</td>
<td>.93</td>
<td>.93</td>
<td>.933</td>
<td>.95</td>
<td>.95</td>
<td>.949</td>
<td>.96</td>
<td>.95</td>
<td>.955</td>
<td>.99</td>
<td>.99</td>
<td>.993</td>
</tr>
<tr>
<td>HIT-SCIR</td>
<td>.93</td>
<td>.93</td>
<td>.926</td>
<td>.93</td>
<td>.93</td>
<td>.930</td>
<td>.95</td>
<td>.95</td>
<td>.953</td>
<td>.99</td>
<td>.99</td>
<td>.993</td>
</tr>
<tr>
<td>SUDA–Alibaba</td>
<td>.91</td>
<td>.91</td>
<td>.911</td>
<td>.90</td>
<td>.91</td>
<td>.903</td>
<td>.91</td>
<td>.92</td>
<td>.915</td>
<td>.97</td>
<td>.99</td>
<td>.982</td>
</tr>
<tr>
<td>Peking</td>
<td>.93</td>
<td>.93</td>
<td>.927</td>
<td>.92</td>
<td>.91</td>
<td>.915</td>
<td>.95</td>
<td>.94</td>
<td>.945</td>
<td>.99</td>
<td>.99</td>
<td>.991</td>
</tr>
</tbody>
</table>

| EDS           |      |           |           |        |           |           |            |           |         |           |       |            |
| ERG           | .90  | .90       | .902      | .97    | .96       | .965      | .96        | .96       | .960    | .96      | .96   | .963      |
| SUDA–Alibaba  | .90  | .90       | .899      | .91    | .91       | .912      | .89        | .91       | .897    | .95      | .95   | .949      |
| HIT-SCIR      | .88  | .82       | .852      | .90    | .89       | .894      | .89        | .91       | .895    | .95      | .94   | .943      |
| SJTU–NICT     | .91  | .85       | .877      | .93    | .86       | .894      | .79        | .76       | .775    | .97      | .90   | .934      |
| Peking        | .83  | .83       | .829      | .95    | .94       | .946      | .91        | .96       | .936    | .96      | .96   | .961      |

|               |      |           |           |        |           |           |            |           |         |           |       |            |
| Peking        | .83  | .83       | .829      | .95    | .94       | .946      | .91        | .96       | .936    | .96      | .96   | .961      |

|               |      |           |           |        |           |           |            |           |         |           |       |            |
| Peking        | .83  | .83       | .829      | .95    | .94       | .946      | .91        | .96       | .936    | .96      | .96   | .961      |
### Comparison to Top-Performing Data-Driven Parsers

<table>
<thead>
<tr>
<th></th>
<th>Tops</th>
<th>Labels</th>
<th>Properties</th>
<th>Anchors</th>
<th>Edges</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>P</td>
<td>R</td>
<td>F&lt;sub&gt;1&lt;/sub&gt;</td>
<td>P</td>
<td>R</td>
</tr>
<tr>
<td>ERG</td>
<td>.92</td>
<td>.92</td>
<td>.918</td>
<td>.99</td>
<td>.99</td>
</tr>
<tr>
<td>SJTU–NICT</td>
<td>.93</td>
<td>.93</td>
<td>.933</td>
<td>.95</td>
<td>.95</td>
</tr>
<tr>
<td>HIT-SCIR</td>
<td>.93</td>
<td>.93</td>
<td>.926</td>
<td>.93</td>
<td>.93</td>
</tr>
<tr>
<td>SUDA–Alibaba</td>
<td>.91</td>
<td>.91</td>
<td>.911</td>
<td>.90</td>
<td>.91</td>
</tr>
<tr>
<td>Peking</td>
<td>.93</td>
<td>.93</td>
<td>.927</td>
<td>.92</td>
<td>.91</td>
</tr>
<tr>
<td>ERG</td>
<td>.90</td>
<td>.90</td>
<td>.902</td>
<td>.97</td>
<td>.96</td>
</tr>
<tr>
<td>SUDA–Alibaba</td>
<td>.90</td>
<td>.90</td>
<td>.899</td>
<td>.91</td>
<td>.91</td>
</tr>
<tr>
<td>HIT-SCIR</td>
<td>.88</td>
<td>.82</td>
<td>.852</td>
<td>.90</td>
<td>.89</td>
</tr>
<tr>
<td>SJTU–NICT</td>
<td>.91</td>
<td>.85</td>
<td>.877</td>
<td>.93</td>
<td>.86</td>
</tr>
<tr>
<td>Peking</td>
<td>.83</td>
<td>.83</td>
<td>.829</td>
<td>.95</td>
<td>.94</td>
</tr>
</tbody>
</table>
### Comparison to Top-Performing Data-Driven Parsers

<table>
<thead>
<tr>
<th></th>
<th>Tops</th>
<th>Labels</th>
<th>Properties</th>
<th>Anchors</th>
<th>Edges</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>P</td>
<td>R</td>
<td>F₁</td>
<td>P</td>
<td>R</td>
</tr>
<tr>
<td><strong>ERG</strong></td>
<td>.92</td>
<td>.92</td>
<td><strong>.918</strong></td>
<td>.99</td>
<td>.99</td>
</tr>
<tr>
<td><strong>SJTU–NICT</strong></td>
<td>.93</td>
<td>.93</td>
<td><strong>.933</strong></td>
<td>.95</td>
<td>.95</td>
</tr>
<tr>
<td><strong>HIT-SCIR</strong></td>
<td>.93</td>
<td>.93</td>
<td><strong>.926</strong></td>
<td>.93</td>
<td>.93</td>
</tr>
<tr>
<td><strong>SUDA–Alibaba</strong></td>
<td>.91</td>
<td>.91</td>
<td><strong>.911</strong></td>
<td>.90</td>
<td>.91</td>
</tr>
<tr>
<td><strong>Peking</strong></td>
<td>.93</td>
<td>.93</td>
<td><strong>.927</strong></td>
<td>.92</td>
<td>.91</td>
</tr>
</tbody>
</table>

#### EDM

<table>
<thead>
<tr>
<th></th>
<th>Tops</th>
<th>Labels</th>
<th>Properties</th>
<th>Anchors</th>
<th>Edges</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>ERG</strong></td>
<td>.90</td>
<td>.90</td>
<td><strong>.902</strong></td>
<td>.97</td>
<td>.96</td>
</tr>
<tr>
<td><strong>SUDA–Alibaba</strong></td>
<td>.90</td>
<td>.90</td>
<td>.899</td>
<td>.91</td>
<td>.91</td>
</tr>
<tr>
<td><strong>HIT-SCIR</strong></td>
<td>.88</td>
<td>.82</td>
<td><strong>.852</strong></td>
<td>.90</td>
<td>.89</td>
</tr>
<tr>
<td><strong>SJTU–NICT</strong></td>
<td>.91</td>
<td>.85</td>
<td><strong>.877</strong></td>
<td>.93</td>
<td>.86</td>
</tr>
<tr>
<td><strong>Peking</strong></td>
<td>.83</td>
<td>.83</td>
<td><strong>.829</strong></td>
<td>.95</td>
<td>.94</td>
</tr>
</tbody>
</table>

Note: Values in bold indicate the best performance in each category.
## Comparison to Top-Performing Data-Driven Parsers

<table>
<thead>
<tr>
<th></th>
<th>Tops</th>
<th>Labels</th>
<th>Properties</th>
<th>Anchors</th>
<th>Edges</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>P  R  F₁</td>
<td>P  R  F₁</td>
<td>P  R  F₁</td>
<td>P  R  F₁</td>
<td>P  R  F₁</td>
</tr>
<tr>
<td>ERG</td>
<td>.92 .92 .918</td>
<td>.99 .99</td>
<td>.987</td>
<td>.96 .96</td>
<td>.956</td>
</tr>
<tr>
<td>SJTU–NICT</td>
<td>.93 .93</td>
<td>.933</td>
<td>.95 .95</td>
<td>.949</td>
<td>.96 .95</td>
</tr>
<tr>
<td>DM</td>
<td>.93 .93 .926</td>
<td>.93 .93</td>
<td>.930</td>
<td>.95 .95</td>
<td>.953</td>
</tr>
<tr>
<td>HIT-SCIR</td>
<td>.91 .91</td>
<td>.911</td>
<td>.90 .91</td>
<td>.903</td>
<td>.91 .92</td>
</tr>
<tr>
<td>SUDA–Alibaba</td>
<td>.91 .91</td>
<td>.915</td>
<td>.95 .94</td>
<td>.945</td>
<td>.99 .99</td>
</tr>
<tr>
<td>Peking</td>
<td>.93 .93</td>
<td>.927</td>
<td>.92 .91</td>
<td>.915</td>
<td>.95 .94</td>
</tr>
<tr>
<td></td>
<td>ERG</td>
<td>.90 .90</td>
<td>.902</td>
<td>.97 .96</td>
<td>.965</td>
</tr>
<tr>
<td>SUDA–Alibaba</td>
<td>.90 .90</td>
<td>.899</td>
<td>.91 .91</td>
<td>.912</td>
<td>.89 .91</td>
</tr>
<tr>
<td>EDS</td>
<td>.88 .82</td>
<td>.852</td>
<td>.90 .89</td>
<td>.894</td>
<td>.89 .91</td>
</tr>
<tr>
<td>HIT-SCIR</td>
<td>.91 .85</td>
<td>.877</td>
<td>.93 .86</td>
<td>.894</td>
<td>.79 .76</td>
</tr>
<tr>
<td>SJTU–NICT</td>
<td>.91 .85</td>
<td>.877</td>
<td>.93 .86</td>
<td>.894</td>
<td>.79 .76</td>
</tr>
<tr>
<td>Peking</td>
<td>.83 .83</td>
<td>.829</td>
<td>.95 .94</td>
<td>.946</td>
<td>.91 .96</td>
</tr>
</tbody>
</table>
### Comparison to Top-Performing Data-Driven Parsers

<table>
<thead>
<tr>
<th></th>
<th>Tops</th>
<th>Labels</th>
<th>Properties</th>
<th>Anchors</th>
<th>Edges</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>P</td>
<td>R</td>
<td>F₁</td>
<td>P</td>
<td>R</td>
</tr>
<tr>
<td>ERG</td>
<td>.92</td>
<td>.92</td>
<td>.918</td>
<td>.99</td>
<td>.99</td>
</tr>
<tr>
<td>SJTU–NICT</td>
<td>.93</td>
<td>.93</td>
<td>.933</td>
<td>.95</td>
<td>.95</td>
</tr>
<tr>
<td>HIT-SCIR</td>
<td>.93</td>
<td>.93</td>
<td>.926</td>
<td>.93</td>
<td>.93</td>
</tr>
<tr>
<td>SUDA–Alibaba</td>
<td>.91</td>
<td>.91</td>
<td>.911</td>
<td>.90</td>
<td>.91</td>
</tr>
<tr>
<td>Peking</td>
<td>.93</td>
<td>.93</td>
<td>.927</td>
<td>.92</td>
<td>.91</td>
</tr>
<tr>
<td>ERG</td>
<td>.90</td>
<td>.90</td>
<td>.902</td>
<td>.97</td>
<td>.96</td>
</tr>
<tr>
<td>SUDA–Alibaba</td>
<td>.90</td>
<td>.90</td>
<td>.899</td>
<td>.91</td>
<td>.91</td>
</tr>
<tr>
<td>HIT-SCIR</td>
<td>.88</td>
<td>.82</td>
<td>.852</td>
<td>.90</td>
<td>.89</td>
</tr>
<tr>
<td>SJTU–NICT</td>
<td>.91</td>
<td>.85</td>
<td>.877</td>
<td>.93</td>
<td>.86</td>
</tr>
<tr>
<td>Peking</td>
<td>.83</td>
<td>.83</td>
<td>.829</td>
<td>.95</td>
<td>.94</td>
</tr>
</tbody>
</table>
A similar technique is almost impossible to apply to other crops.

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


SJTU-NICT at MRP 2019: Multi-Task Learning for End-to-End Uniform Semantic Graph Parsing

Zuchao Li$^{1,2,3}$, Hai Zhao$^{1,2,3}$,* Zhuosheng Zhang$^{1,2,3}$, Rui Wang$^{4,*}$, Masao Utiyama$^4$, and Eiichiro Sumita$^4$

$^1$Department of Computer Science and Engineering, Shanghai Jiao Tong University (SJTU)  
$^2$Key Laboratory of Shanghai Education Commission for Intelligent Interaction and Cognitive Engineering, Shanghai Jiao Tong University, Shanghai, China  
$^3$MoE Key Lab of Artificial Intelligence, AI Institute, Shanghai Jiao Tong University, China  
$^4$National Institute of Information and Communications Technology (NICT), Kyoto, Japan

Abstract

This paper describes our SJTU-NICT’s system for participating in the shared task on Cross-Framework Meaning Representation Parsing (MRP) at the 2019 Conference for Computational Language Learning (CoNLL). Our system uses a graph-based approach to model a variety of semantic graph parsing tasks. Our main contributions in the submitted system are summarized as follows: 1. Our model is fully end-to-end and is capable of being trained only on the given training set which does not rely on any other extra training source including the companion data provided by the organizer; 2. We extend our graph pruning algorithm to a variety of semantic graphs, including the companion data provided by the organizer; and 3. We introduce multi-task learning for multiple objectives within the graph search space; 3. We introduce multi-task learning for multiple objectives within the graph search space. However, due to the variety of semantic graph flavors, the framework-specific “balkanization” of semantic parsing is worth noting. The 2019 Conference on Computational Language Learning (CoNLL) hosts a shared task on Cross-Framework Meaning Representation Parsing (MRP 2019) (Oepen et al., 2019). From the perspective of the formal representation of semantic graphs, MRP 2019 uses the directed graphs to unify the five different semantic representation frameworks: DELPH-IN MRS Bi-Lexical Dependencies (DM), Prague Semantic Dependencies (PSD), Elementary Dependency
Remote Presentation

ShanghaiTech at MRP 2019: Sequence-to-Graph Transduction with Second-Order Edge Inference for Cross-Framework Meaning Representation Parsing

Xinyu Wang, Yixian Liu, Zixia Jia, Chengyue Jiang, Kewe Tu
School of Information Science and Technology, ShanghaiTech University, Shanghai, China
{wangxyl, liuyx, jiazx, jiangchy, tukw}@shanghaitech.edu.cn

Abstract

This paper presents the system used in our submission to the CoNLL 2019 shared task: Cross-Framework Meaning Representation Parsing. Our system is a graph-based parser which combines an extended pointer-generator network that generates nodes and a second-order mean field variational inference module that predicts edges. Our system achieved 1st and 2nd place for the DM and PSD frameworks respectively on the in-framework ranks and achieved 3rd place for the DM framework on the cross-framework ranks. The shared task also provides a cross-framework metric which evaluates the similarity of graph components in all frameworks.
Compositional Parsing Across All Graphbanks

Saarland at MRP 2019
L. Donatelli, M. Fowlie, J. Groschwitz, A. Koller, M. Lindemann, M. Mina, P. Weißenhorn

• Compositional neural parser with competitive results across all MRP shared task graphbanks (only compositional parser to do so!)
  • 4th place overall
  • 1st on PSD
  • 1st The Little Prince subset

• Parser previously held SOTA on MRP graphbanks apart from UCCA at ACL 2019
Apply-Modify (AM) Algebra and graph decomposition

1. Input: The tall giraffe wants to eat
2. Neural supertagging + dependency parsing
3. AM dependency tree
4. Evaluates deterministically
5. Output

- Linguistically-motivated compositional structure
- Diverse meaning representations mapped to similar AM trees
HIT-SCIR at MRP 2019:  
A Unified Pipeline for Meaning Representation Parsing via Efficient Training and Effective Encoding

Wanxiang Che, Longxu Dou, Yang Xu, Yuxuan Wang, Yijia Liu, Ting Liu
Research Center for Social Computing and Information Retrieval
Harbin Institute of Technology
Overview of Our Techniques

• Rank 1\textsuperscript{st} according to ALL-F1
• Baseline model: Transition-based Parser with Stack LSTM (Dyer et al., 2015)
• Our Extensions:
  • Efficient Training of Stack LSTM via parallel training
  • Effective Encoding via adopting BERT

<table>
<thead>
<tr>
<th>System</th>
<th>DM</th>
<th>PSD</th>
<th>EDS</th>
<th>UCCA</th>
<th>AMR</th>
<th>ALL-F1</th>
</tr>
</thead>
<tbody>
<tr>
<td>HIT-SCIR</td>
<td>95.08</td>
<td>90.55</td>
<td>90.75</td>
<td>81.67</td>
<td>72.94</td>
<td>86.2</td>
</tr>
<tr>
<td>SJTU-NICT</td>
<td>95.50</td>
<td>91.19</td>
<td>89.90</td>
<td>77.80</td>
<td>71.97</td>
<td>85.3</td>
</tr>
<tr>
<td>Suda-Alibaba</td>
<td>92.26</td>
<td>85.56</td>
<td>\textbf{91.85}</td>
<td>78.43</td>
<td>71.72</td>
<td>84.0</td>
</tr>
<tr>
<td>Saarland</td>
<td>94.69</td>
<td>91.28</td>
<td>89.10</td>
<td>67.55</td>
<td>66.72</td>
<td>81.9</td>
</tr>
<tr>
<td>Hitachi</td>
<td>91.02</td>
<td>91.21</td>
<td>83.74</td>
<td>70.36</td>
<td>43.86</td>
<td>76.0</td>
</tr>
<tr>
<td>Amazon</td>
<td>93.26</td>
<td>89.98</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>\textbf{73.38}</td>
</tr>
</tbody>
</table>
Parallel Training Stack-LSTM

- Aligning the similar operations in Stack-LSTM within a batch
- Computing them simultaneously

- Conduct experiments with GloVe
- 5.3x on DM
- 2.7x on UCCA
BERT is Amazing!

- We fine-tune the BERT
- Layer-wise scalar weighed BERT is adopted

<table>
<thead>
<tr>
<th>Feature</th>
<th>DM</th>
<th>PSD</th>
<th>EDS</th>
<th>UCCA</th>
<th>AMR</th>
<th>Avg</th>
</tr>
</thead>
<tbody>
<tr>
<td>GloVe</td>
<td>87.1</td>
<td>74.1</td>
<td>88.2</td>
<td>87.5</td>
<td>65.3</td>
<td>80.4</td>
</tr>
<tr>
<td>BERT(base)</td>
<td>94.3</td>
<td>83.6</td>
<td>91.5</td>
<td>92.8</td>
<td>71.4</td>
<td>86.7</td>
</tr>
</tbody>
</table>

- Metric: ALL-F1 based on mtool
- Dataset: Splited training data on 8:1:1 proportion
Structure vs Representation

- Transition-based Parser achieves comparable results with Graph-based Parser
- Kulmizev et al. (2019) found similar conclusion in PTB

<table>
<thead>
<tr>
<th>Model</th>
<th>Feature</th>
<th>DM</th>
<th>PAS</th>
<th>PSD</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>id</td>
<td>ood</td>
<td>id</td>
</tr>
<tr>
<td>Wang et al</td>
<td>word2vec</td>
<td>89.3</td>
<td>83.2</td>
<td>91.4</td>
</tr>
<tr>
<td>Dozat et al</td>
<td>GloVe+char</td>
<td>92.7</td>
<td>87.8</td>
<td>94.0</td>
</tr>
<tr>
<td>Transition</td>
<td>GloVe+char</td>
<td>86.1</td>
<td>79.2</td>
<td>89.8</td>
</tr>
<tr>
<td>Graph</td>
<td>GloVe+char</td>
<td>91.6</td>
<td>86.1</td>
<td>93.1</td>
</tr>
<tr>
<td>Transition</td>
<td>BERT</td>
<td>92.9</td>
<td>89.2</td>
<td>94.4</td>
</tr>
<tr>
<td>Graph</td>
<td>BERT</td>
<td><strong>94.1</strong></td>
<td><strong>90.8</strong></td>
<td><strong>94.8</strong></td>
</tr>
</tbody>
</table>

Wang et al: <A Neural Transition-Based Approach for Semantic Dependency Graph Parsing>
Dozat et al: <Simpler but More Accurate Semantic Dependency Parsing>
Kulmizev et al: <Deep Contextualized Word Embeddings in Transition-Based and Graph-Based Dependency Parsing – A Tale of Two Parsers Revisited>
Model Ensemble

• In follow up experiment, we obtain further improvement on lpps dataset
• Ensemble model consists of 5 single model

<table>
<thead>
<tr>
<th>Systems</th>
<th>DM</th>
<th>PSD</th>
<th>EDS</th>
<th>UCCA</th>
<th>AMR</th>
<th>Avg</th>
</tr>
</thead>
<tbody>
<tr>
<td>Single</td>
<td>93.98</td>
<td>87.41</td>
<td>89.83</td>
<td>82.61</td>
<td>69.03</td>
<td>84.57</td>
</tr>
<tr>
<td>Ensemble</td>
<td>94.00</td>
<td>87.79</td>
<td>89.57</td>
<td>83.41</td>
<td>71.35</td>
<td>85.16</td>
</tr>
</tbody>
</table>
Conclusion

• Our Contribution:
  • Efficient Training of Stack LSTM via parallel training
  • Effective Encoding through adopting BERT

• The performance gap between Graph and Transition on SDP is almost eliminated under BERT

• Our code: https://github.com/HIT-SCIR/HIT-SCIR-CoNLL2019
SJTU at MRP 2019: A Transition-Based Multi-Task Parser for Cross-Framework Meaning Representation Parsing

Hongxiao Bai\textsuperscript{1,2,3}, Hai Zhao\textsuperscript{1,2,3,*}

\textsuperscript{1}Department of Computer Science and Engineering, Shanghai Jiao Tong University
\textsuperscript{2}Key Laboratory of Shanghai Education Commission for Intelligent Interaction and Cognitive Engineering, Shanghai Jiao Tong University, Shanghai, China
\textsuperscript{3}MoE Key Lab of Artificial Intelligence, AI Institute, Shanghai Jiao Tong University

\texttt{baippa@sjtu.edu.cn, zhaohai@cs.sjtu.edu.cn}

Abstract

This paper describes the system of our team \textit{SJTU} for our participation in the CoNLL 2019 Shared Task: Cross-Framework Meaning Representation Parsing. The goal of the task is to advance data-driven parsing into graph-structured representations of sentence meaning. This task includes five meaning representation frameworks: DM, PSD, EDS, UCCA, and AMR. These frameworks have different properties and structures. To tackle all the frameworks in one model, it is needed to find out the commonality of them. In our work, we define a set of the transition actions to once-

shallow syntax and in particular for representations of the semantic structure. Many works have shown that these meaning representations are beneficial to other tasks such as machine translation and abstractive summarization. However, there are several types of meaning representations with different definitions, structures, and abstractions, which hinder the applications.

The CoNLL 2019 Shared Task (Oepen et al., 2019) combines formally and linguistically different meaning representation in graph form on a uniform training and evaluation setup for the first time. This task includes five MRP frame-
JBNNU at MRP 2019: Multi-level Biaffine Attention for Semantic Dependency Parsing

Seung-Hoon Na, Jinwoo Min, Kwanghyeon Park
Dept. Computer Science, Jeonbuk National University

Jong-Hun Shin, Young-Kil Kim
Electronics and Telecommunication Research Institute (ETRI)
Introduction

• Our issue: **Multi-task learning** for **DM/PSD/UCCA**
  
  – To enable **multi-task learning**, we explicitly make **shared common components** in a neural network architecture across different frameworks

• Models
  
  – **Biaffine attention**: we propose a unified neural model for the **DM/PSD/UCCA** frameworks based on the biaffine attention [Dozat and Manning, 2017, 2018; Zhang et al., 2019]

  – **Multi-level biaffine attention**:
    
    • Motivated by the multi-level architecture of FusionNet in the machine reading comprehension task [Huang et al., 2018]
This technique is impossible to apply...

Encoder: BERT-BiLSTM ← shared across frameworks

Decoder: Biaffine attention ← framework specific
**Encoder: BERT-BiLSTM (shared across frameworks)**

- **Word representation layer using BERT**

  Given a sentence, the BERT encoder is applied to its wordpieces and the encoded wordpiece-level representations are composed to the word-level embeddings based on BiLSTM

\[
x_i = \left[ w_{i}^{bert}; e_{i}^{glove}; e_{i}^{POS} \right]
\]

\[
r_{i} = BiLSTM_{i}(x_{1} \cdots x_{n})
\]
Decoder: Biaffine attention *(framework specific)*

- Biaffine attention is performed on the role-dependent representations to predict the existence of an edge and its labels.

### Biaffine attention

\[
BiAff_m(x, y) = x^T U^{[1:m]} y + V \begin{bmatrix} x \\ y \end{bmatrix} + b
\]

\[
s_{i,j}^{(edge)} = BiAff_{1}^{(edge)} \left( h_{i}^{(dep)}, h_{j}^{(head)} \right)
\]

\[
s_{i,j}^{(label)} = BiAff_{k}^{(label)} \left( h_{i}^{(l-dep)}, h_{j}^{(l-head)} \right)
\]

\[
s_{i}^{(top)} = FFN^{(top)} (r_{i})
\]
Multi-level Biaffine attention

The hidden representations at three levels are composed to the final hidden representation $z_i^{(dep)}, z_i^{(head)}$ using a semantic fusion unit.
Preliminary Experiment

For more details, please visit our poster. Thank you.

<table>
<thead>
<tr>
<th>method</th>
<th>DM</th>
<th>PSD</th>
<th>UCCA</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Top</td>
<td>UF</td>
<td>LF</td>
</tr>
<tr>
<td>Biaffine</td>
<td>93.67</td>
<td>92.08</td>
<td>90.86</td>
</tr>
<tr>
<td>BERT+Biaffine</td>
<td>95.06</td>
<td>93.85</td>
<td>93.00</td>
</tr>
<tr>
<td>BERT+Multi-level Biaffine</td>
<td>95.09</td>
<td>93.86</td>
<td>93.02</td>
</tr>
<tr>
<td>BERT+Biaffine+MTL</td>
<td>N/A</td>
<td>93.66</td>
<td>92.73</td>
</tr>
</tbody>
</table>

- **BERT+Biaffine** performs better than Biaffine, in particular, obtaining the increases of about 5% for UF and LF on the UCCA framework.
- **BERT+Multi-level Biaffine** does not achieve any further improvements with respect to BERT-Biaffine model.
- **BERT+Biaffine+MTL** only achieves small improvements on UCCA framework whereas no improvements on DM and PSD frameworks can be observed.
CUHK at MRP2019: Transition-Based Parser with Cross-Framework Variable-Arity Resolve Action

Sunny Lai  Chun Hei Lo  Kwong Sak Leung  Yee Leung
Abstract

Our system:
- **transition-based** parser
- **directed acyclic graph (DAG)** to tree preprocessor
- **cross-framework variable-arity** RESOLVE action that generalizes over five different representations.
- Although we ranked low in the competition, we have shown the current limitations and potentials of including variable-arity action in MRP and concluded with directions for improvements in the future.
Abstract

Our system:
- transition-based parser
- directed acyclic graph (DAG) to tree preprocessor
- cross-framework variable-arity RESOLVE action that generalizes over five different representations.
- Although we ranked low in the competition, we have shown the current limitations and potentials of including variable-arity action in MRP and concluded with directions for improvements in the future.

Arity: is the number of arguments or operands (No. of nodes) that the function takes (Wikipedia)

Standard shift reduce: 2, This paper: n
# Motivation

<table>
<thead>
<tr>
<th>MRP</th>
<th>F</th>
<th>Actions</th>
<th>Author</th>
</tr>
</thead>
<tbody>
<tr>
<td>PSD</td>
<td>0</td>
<td>LEFT-REDUCE(L), RIGHT-SHIFT(L), NO-SHIFT, NO-REDUCE, LEFT-PASS(L), RIGHT-PASS(L), NO-PASS</td>
<td>(Wang et al., 2018)</td>
</tr>
<tr>
<td>UCCA</td>
<td>1</td>
<td>SHIFT, REDUCE, <strong>NODE(X)</strong>, LEFT-EDGE(X), RIGHT-EDGE(X), LEFT-REMOTE(X), RIGHT-REMOTE(X), SWAP, FINISH</td>
<td>(Hershcovich et al., 2017)</td>
</tr>
<tr>
<td>AMR</td>
<td>2</td>
<td>SHIFT, REDUCE, RIGHT-LABEL(R), LEFT-LABEL(R), SWAP, MERGE, PRED(N), <strong>ENTITY(L)</strong>, GEN(N)</td>
<td>(Guo and Lu, 2018)</td>
</tr>
<tr>
<td>*</td>
<td>*</td>
<td><strong>SHIFT</strong>, <strong>IGNORE</strong>, <strong>RESOLVE</strong></td>
<td>This paper</td>
</tr>
</tbody>
</table>

We introduce the cross-framework variable-arity **RESOLVE** action as:

1. there is **no need** to include additional **binarization** of the dependencies and reduce the number of transitions
2. It is also more natural to **consider the dependency of multiple nodes jointly** as meaning representations like semantic frames usually involve multiple arguments
3. Learn **cross-framework features to generalize** our model
## Motivation

<table>
<thead>
<tr>
<th>MRP</th>
<th>F</th>
<th>Actions</th>
<th>Author</th>
</tr>
</thead>
<tbody>
<tr>
<td>PSD</td>
<td>0</td>
<td>LEFT-REDUCE(L), RIGHT-SHIFT(L), NO-SHIFT, NO-REDUCE, LEFT-PASS(L), RIGHT-PASS(L), NO-PASS</td>
<td>(Wang et al., 2018)</td>
</tr>
<tr>
<td>UCCA</td>
<td>1</td>
<td>SHIFT, REDUCE, NODE(X), LEFT-EDGE(X), RIGHT-EDGE(X), LEFT-REMOTE(X), RIGHT-REMOTE(X), SWAP, FINISH</td>
<td>(Hershcovich et al., 2017)</td>
</tr>
<tr>
<td>AMR</td>
<td>2</td>
<td>SHIFT, REDUCE, RIGHT-LABEL(R), LEFT-LABEL(R), SWAP, MERGE, PRED(N), ENTITY(L), GEN(N)</td>
<td>(Guo and Lu, 2018)</td>
</tr>
</tbody>
</table>

* * * **SHIFT, IGNORE, RESOLVE**  

This paper

We introduce the cross-framework variable-arity **RESOLVE** action as:

1. **no need** to include additional **binarization** of the dependencies and reduce the number of transitions
2. It is also more natural to **consider the dependency of multiple nodes jointly** as meaning representations like semantic frames usually involve multiple arguments
3. Learn **cross-framework features to generalize** our model
Approach

Top-node oriented tree

DM

MRP

Top-node

Initial tokenized nodes queue: [A, full, , four-color, page, in, Newsweek, will, cost, $, 100,980]
Approach

Top-node oriented tree

DM

MRP

Top-node

Initial tokenized nodes queue: [A, full, , four-color, page, in, Newsweek, will, cost, $, 100,980]
Approach

Top-node oriented tree

Initial tokenized nodes queue: [A, full, , four-color, page, in, Newsweek, will, cost, $, 100,980]
Approach

Top-node oriented tree

DM

MRP

Initial tokenized nodes queue: [A, full, ,, four-color, page, in, Newsweek, will, cost, $, 100,980]
A node can be **RESOLVED** when all its child is **RESOLVED**

**RESOLVE:**
- Predict node labels (framework specific)
- Build edges (framework specific)

Initial tokenized nodes queue: [A, full, , four-color, page, in, Newsweek, will, cost, $, 100,980]
Approach

A node can be **RESOLVED** when all its child is **RESOLVED**

**RESOLVE:**
- Predict node labels (framework specific)
- Build edges (framework specific)

<table>
<thead>
<tr>
<th>Action</th>
<th>n</th>
<th>Stack</th>
</tr>
</thead>
<tbody>
<tr>
<td>SHIFT</td>
<td></td>
<td>[A]</td>
</tr>
<tr>
<td>RESOLVE</td>
<td>1</td>
<td>[a]</td>
</tr>
</tbody>
</table>

Initial tokenized nodes queue: `[A, full, , four-color, page, in, Newsweek, will, cost, $, 100,980]`
A node can be **RESOLVED** when all its child is **RESOLVED**

The approach involves building a top-node oriented tree with specific actions:

- Predict node labels (framework specific)
- Build edges (framework specific)

### RESOLVE:
- Predict node labels (framework specific)
- Build edges (framework specific)

<table>
<thead>
<tr>
<th>Action</th>
<th>n</th>
<th>Stack</th>
</tr>
</thead>
<tbody>
<tr>
<td>[a, full, color, page, in]</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**RESOLVE** 5 [page]

- page → a, page → full
- page → color, page → in

Initial tokenized nodes queue: [A, full, , four-color, page, in, Newsweek, will, cost, $, 100,980]
<table>
<thead>
<tr>
<th>Submission</th>
<th>tops</th>
<th>labels</th>
<th>properties</th>
<th>anchors</th>
<th>edges</th>
<th>attributes</th>
<th>all</th>
</tr>
</thead>
<tbody>
<tr>
<td>TUPA(multi)</td>
<td>0.616</td>
<td>0.457</td>
<td>0.327</td>
<td>0.626</td>
<td>0.347</td>
<td>0.037</td>
<td>0.453</td>
</tr>
<tr>
<td>RESOLVER</td>
<td>0.502</td>
<td>0.365</td>
<td>0.317</td>
<td>0.568</td>
<td>0.095</td>
<td>0.00</td>
<td>0.378</td>
</tr>
</tbody>
</table>

- Cross-framework variable-arity actions are **hard to learn**
- **Information loss** happens when converting graphs to tree structures.
- **Model design** can still be improved.
Acknowledgement

• Organizers (especially Stephan Oepen)
• Anonymous Reviewers
• Teammates (especially Aaron Lo)
• Other participating teams (especially Longxu Dou)

We welcome any comments and suggestions!
slai@cse.cuhk.edu.hk
Our Approach

- Unify graph predictions with a single encoder-to-biaffine network
- Multi-task variant of the unified system (in post evaluation)

1. Extract task-independent contextualized token representations with shared encoder
2. Complement missing nodes
3. Predict edges and their labels with biaffine networks [Dozat+18]
## Frame Specific Approaches

<table>
<thead>
<tr>
<th>Framework</th>
<th>Biaffine Like Net.</th>
<th>Rule</th>
<th>Linear model</th>
<th>Generator</th>
</tr>
</thead>
<tbody>
<tr>
<td>DM and PSD</td>
<td>Edge + Frame</td>
<td>Properties</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>EDS</td>
<td>Node anchor</td>
<td>Node &amp; Edge gen.</td>
<td>Node &amp; Edge gen.</td>
<td>-</td>
</tr>
<tr>
<td>UCCA</td>
<td>Edge</td>
<td>-</td>
<td>-</td>
<td>Non-terminal node: pointer network</td>
</tr>
<tr>
<td>AMR</td>
<td>Edge</td>
<td>Preprocess + serialize</td>
<td>-</td>
<td>Node: pointer-generator network</td>
</tr>
</tbody>
</table>

## Results

<table>
<thead>
<tr>
<th>Variant</th>
<th>Average</th>
<th>DM</th>
<th>PSD</th>
<th>EDS</th>
<th>UCCA</th>
<th>AMR</th>
</tr>
</thead>
<tbody>
<tr>
<td>SFL</td>
<td>0.7575/5</td>
<td>0.9071/9</td>
<td>0.9064/3</td>
<td>0.8339/7</td>
<td>0.7014/6</td>
<td>0.4386/8</td>
</tr>
<tr>
<td>SFL(ensemble)*</td>
<td>0.7604/5</td>
<td>0.9102/9</td>
<td>0.9121/2</td>
<td>0.8374/7</td>
<td>0.7036/6</td>
<td>0.4386/8</td>
</tr>
<tr>
<td>BERT+SFL(NT)</td>
<td>0.7450/6</td>
<td>0.9038/9</td>
<td>0.9069/3</td>
<td>0.8301/7</td>
<td>0.6945/6</td>
<td>0.3896/8</td>
</tr>
<tr>
<td>BERT+MTL(NT)</td>
<td>0.7144/6</td>
<td>0.8726/9</td>
<td>0.8791/7</td>
<td>0.7987/7</td>
<td>0.6422/6</td>
<td>0.3794/9</td>
</tr>
<tr>
<td>BERT+MTL+FT(NT)</td>
<td>0.7507/5</td>
<td>0.9045/9</td>
<td>0.9054/4</td>
<td>0.8304/7</td>
<td>0.7126/6</td>
<td>0.4008/8</td>
</tr>
</tbody>
</table>
ÚFAL MRPipe at MRP2019: UDPipe Goes Semantic in the Meaning Representation Parsing Shared Task

Milan Straka, Jana Straková

November 3, 2019
MRPipe Design

- start completely from scratch
MRPipe Design

- start completely from scratch

- uniform architecture for simple directed graph parsing
  - i.e., we consider at most one edge for all pairs of nodes in both directions
  - therefore, we can model trees, DAGs, even cycles
  - we could model hypergraphs (i.e., parallel edges) easily, but we did not yet evaluated it (~0.4% parallel edges in AMR, ~1.25% in UCCA)
MRPipe Design

- start completely from scratch
- uniform architecture for simple directed graph parsing
  - i.e., we consider at most one edge for all pairs of nodes in both directions
  - therefore, we can model trees, DAGs, even cycles
  - we could model hypergraphs (i.e., parallel edges) easily, but we did not yet evaluated it (~0.4% parallel edges in AMR, ~1.25% in UCCA)
- no linguistic information, structural constraints, dicts, ...
MRPipe Design

- start completely from scratch
- uniform architecture for simple directed graph parsing
  - i.e., we consider at most one edge for all pairs of nodes in both directions
  - therefore, we can model trees, DAGs, even cycles
  - we could model hypergraphs (i.e., parallel edges) easily, but we did not yet evaluated it (~0.4% parallel edges in AMR, ~1.25% in UCCA)
- no linguistic information, structural constraints, dicts, ...
- rich pretrained embeddings – frozen BERT embeddings
MRPipe Parsing Algorithm

- consider **tokens** as **nodes**, **anchors** as **edges** to them
MRPipe Parsing Algorithm

- consider **tokens** as **nodes**, **anchors** as **edges** to them
- construct the graph layerwise by interleaving following two operations:
MRPipe Parsing Algorithm

- consider **tokens** as **nodes**, **anchors** as **edges** to them
- construct the graph layerwise by interleaving following two operations:
  - **AddNodes**: for every node, independently consider creating its new child or parent, with all its properties
MRPipe Parsing Algorithm

- consider **tokens** as **nodes**, **anchors** as **edges** to them
- construct the graph layerwise by interleaving following two operations:
  - **AddNodes**: for every node, independently consider creating its new child or parent, with all its properties
  - **AddEdges**: for every created node, independently consider connecting it to every other node (existing or new), generating all attributes if required
MRPipe Node Representation

- each node is represented as
  - the **underlying token** which generated it (recursively)
  - embeddings of all node **properties**
  - embeddings of all **adjacent** edges **attributes**
MRPipe Node Representation

- each node is represented as
  - the **underlying token** which generated it (recursively)
  - embeddings of all node **properties**
  - embeddings of all **adjacent** edges **attributes**

- node properties can be encoded **relatively**
  - with respect to the **anchored tokens**
MRPipe Node Representation

- each node is represented as
  - the **underlying token** which generated it (recursively)
  - embeddings of all node **properties**
  - embeddings of all **adjacent** edges **attributes**

- node properties can be encoded **relatively**
  - with respect to the **anchored tokens**
  - automatically choosing **absolute** (e.g., POS, frames) or **relative** encoding (e.g., labels, cargs, op[1-9], ARG[1-9])
MRPipe Example

(a) Left: Initial configuration with tokens only. Right: Token representation encoder architecture.
(b) Left: First **AddNodes** operation. Right: Architecture of the new node classifier and representation encoder.
MRPipe Example

(c) Left: First **AddEdges** operation. Right: Architecture of the edge classifier and updated node representation encoder.
(d) Left: Second **AddNodes** operation. Right: Architecture of the new node classifier and representation encoder.
(e) Left: Second **AddEdges** operation. Right: Architecture of the edge classifier and updated node representation encoder.
MRPipe Results

- we utilized incorrect companion test data for three treebanks (the ones without anchors)
- our fixed submission ranked on a shared 3rd place
- best overall labels and properties scores, worse edges

<table>
<thead>
<tr>
<th>System</th>
<th>Tops</th>
<th>Labels</th>
<th>Properties</th>
<th>Anchors</th>
<th>Edges</th>
<th>Attributes</th>
<th>All</th>
</tr>
</thead>
<tbody>
<tr>
<td>Original ST submission</td>
<td>75.12%</td>
<td>63.99%</td>
<td>56.53%</td>
<td>69.53%</td>
<td>62.17%</td>
<td>7.85%</td>
<td>74.74%</td>
</tr>
<tr>
<td>Bugfix ST submission</td>
<td>81.47%</td>
<td>73.06%</td>
<td>69.95%</td>
<td>77.23%</td>
<td>73.89%</td>
<td>7.87%</td>
<td>83.96%</td>
</tr>
<tr>
<td>99% training data</td>
<td>80.59%</td>
<td>73.06%</td>
<td>70.18%</td>
<td>77.35%</td>
<td>74.27%</td>
<td>7.96%</td>
<td>84.14%</td>
</tr>
<tr>
<td>No BERT embeddings</td>
<td>70.50%</td>
<td>70.71%</td>
<td>67.01%</td>
<td>76.02%</td>
<td>65.02%</td>
<td>5.30%</td>
<td>78.99%</td>
</tr>
<tr>
<td>Ensemble</td>
<td>81.13%</td>
<td>73.39%</td>
<td>70.82%</td>
<td>77.57%</td>
<td>75.85%</td>
<td>8.28%</td>
<td>85.05%</td>
</tr>
<tr>
<td>HIT-SCIR (Che et al., 2019)</td>
<td>90.41%</td>
<td>70.85%</td>
<td>69.86%</td>
<td>77.61%</td>
<td>79.37%</td>
<td>12.40%</td>
<td>86.20%</td>
</tr>
<tr>
<td>SJTU–NICT (Li et al., 2019)</td>
<td>91.50%</td>
<td>71.24%</td>
<td>68.73%</td>
<td>77.62%</td>
<td>77.74%</td>
<td>9.40%</td>
<td>85.27%</td>
</tr>
<tr>
<td>SUDA–Alibaba (Zhang et al., 2019b)</td>
<td>86.01%</td>
<td>69.50%</td>
<td>68.24%</td>
<td>77.11%</td>
<td>76.85%</td>
<td>8.16%</td>
<td>83.96%</td>
</tr>
<tr>
<td>Saarland (Donatelli et al., 2019)</td>
<td>86.70%</td>
<td>71.33%</td>
<td>61.11%</td>
<td>75.08%</td>
<td>75.01%</td>
<td>—</td>
<td>81.87%</td>
</tr>
</tbody>
</table>
MRPipe Future Work

- allow anchoring to sub-token by adding attributes with character indices
- generate nodes one-by-one so that they are conditioned on already generated ones (important for constituency structure)
- better node representation
- better architecture of edge generation (not an independent decision for every edge)
Amazon at MRP 2019: Parsing Meaning Representations with Lexical and Phrasal Anchoring

Jie Cao†*, Yi Zhang‡, Adel Youssef‡, Vivek Srikumar‡
†School of Computing, University of Utah
‡AWS AI, Amazon
{jcao, svivek}@cs.utah.edu, {yizhngn, adel}@amazon.com

Abstract

This paper describes the system submission of our team Amazon to the shared task on Cross Framework Meaning Representation Parsing (MRP) at the 2019 Conference for Computational Language Learning (CoNLL). Via extensive analysis of implicit alignments in AMR, we recategorize five meaning representations (MRs) into two classes: Lexical-Anchoring and Phrasal-Anchoring. Then we propose a unified graph-based parsing framework for the lexical-anchoring MRs, and a phrase-structure parsing for one of the phrasal-anchoring MRs, UCCA. Our system submission ranked 1st in the AMR subtask, and sometimes also with assumptions on underlying syntactic structures.

Anchoring is crucial in graph-based meaning representation parsing. Training a statistical parser typically starts with a conjectured alignment between tokens/spans and the semantic graph nodes to help to factorize the supervision of graph structure into nodes and edges. In our paper, with evidence from previous research on AMR alignments (Pourdamghani et al., 2014; Flanigan et al., 2014; Wang and Xue, 2017; Chen and Palmer, 2017; Szubert et al., 2018; Lyu and Titov, 2018), we propose a uniform handling of three meaning representations from Flavor-0 (DM, PSD) and
SUDA–Alibaba at MRP 2019: Graph-Based Models with BERT

Yue Zhang¹, Wei Jiang², Qingrong Xia²,
Junjie Cao¹, Rui Wang¹, Zhenghua Li²*, Min Zhang²
¹ Alibaba Group, China
² School of Computer Science and Technology, Soochow University, China

poster id #14
**EDS**

- **Node prediction:** original node & append node
  - **Original nodes:** take coarse POS tag and sense as joint label
  - **Append nodes:** predict begin/end index (anchor) and type for each node

- **Edge prediction:**
  - Compute the score of each edge relation between two nodes by the **biaffine** scorer

---

**Append nodes:**

- `compound`

**Sentence:**

Pension Reserves : Holdings by pension funds

**Original nodes:**

- `n_l`
- `n_l`
- `v_id`
- `n_l`
- `-p`
- `n_l`
- `n_l`
- `n_l`
- `O`

**Append nodes:**

- `udf_q`
- `udf_q`
- `udf_q`
- `udf_q`
UCCA

- Convert UCCA to **constituent tree** (Jiang et al. 2019)
  - Remove remote edges
  - Handle discontinuous nodes
- Utilize minimal **span-based parser** (Stern et al. 2017)
- Remote recovery as a new task (multi-task learning)
Welcome to our poster for more details about our work!

SUDA-Alibaba at MRP 2019: Graph-Based Models with BERT
Yue Zhang1, Wei Jiang2, Congcong Xie3, Junjie Cao1, Rui Wang1, Zhenghua Li1, Min Zhang2
1. China University of Petroleum, Beijing, China
2. Nanjing University, Nanjing, China
3. Alibaba Group, Hangzhou, China

Overview
> Our participating systems for the five frameworks can be characterized in the following aspects:
> Graph-Based: We directly predict edges among nodes, instead of using a transition system.
> Joint Model: Sharing the encoder component under the MTL framework and it is adopted by the DM, PSD, and UCCA models.
> BERT: Using BERT as our extra inputs is effective for all the models, except AMR.

EDS
> Convert into two subtasks: node prediction & edge prediction.
> Node prediction: as sequence labeling task.
  - Split nodes into two types: original node and append node according to whether its label begins with 'A' or 'O'.
  - Original nodes (yellow nodes): take coarse POS tag and sense of each node as joint label, predict it for each word.
  - Append nodes (purpose and period nodes): predict begin index, end index (or blank) and type for each node, like SRL predicte and argument prediction corresponding.

UCCA
> Graph-based UCCA parser (Jiang et al. 2019)
  - Convert UCCA graphs to constituent trees.
  - Step 1: remove edges
  - Step 2: handle discontinuous structures
  - Employ minimal-span based parser (Stent et al. 2017)
  - Use shared structure and MTL for recovering remote edges

AMR
> Joint model alignments, concepts and relations (Yu and Tsau, 2018)
  - The concept identification model chooses a concept conditioned on the aligned word based on the BS/STM state.
  - The relation identification model employs a log-linear module with bilinear scorer to compute the probabilities of each concept pair.
  - The alignment model is only used in training.

Our work
> Convert the MRP AMR test to original AMR test.
> Generate the PoS and NER tags with the NER taggers in the white list.

Main Reference
> Timothy Dopp and Christopher D. Manning, 2018. Simpler but more accurate constituent dependency parsing
> Wei Jiang, Zhenhui Li, Yu Zhang, and Min Zhang, 2019. Efficiency of removal of non-terminals in graph parsing as constituency dependency parsing
> Zhenhui Li and Jun Xu, 2018. Amr parsing as graph prediction
> Jiecao Diao, Ming-Wei Chang, Kerstin Lee, and Kristina Toutanova, 2018. BERT: pre-training of deep bidirectional transformers for language understanding

Results on provided test data
> Our overall result on all five tasks ranks third, and our results ranks first on EDS and second on UCCA.
ÚFAL–Oslo at MRP 2019: Garage Sale Semantic Parsing

Kira Droganova,† Andrey Kutuzov,‡
Nikita Mediankin† and Daniel Zeman†

†Charles University, Faculty of Mathematics and Physics, ÚFAL
‡University of Oslo, Faculty of Mathematics and Natural Sciences, Language Technology Group

†{droganova|mediankin|zeman}@ufal.mff.cuni.cz
‡andreku@ifi.uio.no
Garage Sale Semantic Parsing


Garage Sale Semantic Parsing

- create forward conversion scripts;
- create training/development splits;
- create/download all accompanying files;
- convert the data and train a model;
- create backwards conversion scripts.
Garage Sale Semantic Parsing
Peking at MRP 2019: Composition- and Factorization-Based Parsing for Elementary Dependency Structures

Yufei Chen, Yajie Ye and Weiwei Sun

Wangxuan Institute of Computer Technology
Peking University

November 3, 2019
English Resource Semantics

(1) a. Every dog chases some white cats.

b. \(\text{some}(y, \text{white}(y) \land \text{cat}(y), \text{every}(x, \text{dog}(x), \text{chase}(e, x, y)))\)

C. \(\text{every}(x, \text{dog}(x), \text{some}(y, \text{white}(y) \land \text{cat}(y), \text{chase}(e, x, y)))\)

A compact graph-based representation of the two readings

```
_every_q
  /   \\
_RSTR/H /     \\
  /     \\
_dog_n_1

_chase_v_1
  /   \\
ARG1/NEQ /     \\
  /     \\
_cat_n_1

_some_q
  /   \\
_RSTR/H /     \\
  /     \\
_white_a_1
```

Variables \((e, x \text{ and } y)\) are implicitly patched to the predicates that treat them as intrinsic variables (\text{chase, dog and cat})

**Elementary Dependency Structures** Removing H, EQ, NEQ, etc.
String-to-graph parsing approaches

- Factorization-based approach
- Composition-based approach
- Transition-based approach
- Translation-based approach
String-to-graph parsing approaches

- Factorization-based approach
- Composition-based approach
- Transition-based approach
- Translation-based approach
Factorization-based approach

Input

Tom wants to go.

Tokenization

Tom / wants / to / go / .

Concept Identification

proposed.q

named *_v_1 ∅ *_v_1 ∅

Tom wants to go .

Relation Detection

Output

proper_q^{0:3}

ARG2

-want_v_1^{4:9}

ARG1

named(“Tom”)^{0:3}

Property Prediction

proposed_q

ARG2

-go_v_1

ARG1

named
Neural models

\[ \text{ScoreEdge}(\text{pron} \leftarrow \text{go}_v_1) \]

2:pronom \_q
1:pron

3:*_v_1

4:*_v_1

arg max

\[ \text{Biaffine} \]

c_1 \quad c_4

r_1

r_4

encoder

encoder

encoder

encoder

He

wants

to

go
Composition-based approach

Some boys want to go

arg1

ARG1

ARG1

NP

Some

NP

NP

some

ARG2

ARG2

ARG1

ARG1

go

want

to

go

VP

VP

NP

boy

NP

x

z

y

y

x

z

x

z

y

want

go

want

go

Some

boys

PEKING UNIVERSITY
Thanks for your attention!
Meaning Representation
Parsing Shared Task

Discussion
Discussion about the MRP task 2019–2020

• Some possible discussion points
  • Evaluation metric(s) – what to avoid? Improvements?
    • One main metric (even if approximate)? Several “equal” metrics (~ several “winners”)
  • Extending the task
    • More languages (within the same frameworks)?
    • Additional frameworks?
    • Same text across frameworks ([mostly] evaluation only)?
  • Time schedule
    • How much time needed for “ingesting” whitelisted resources?
      • Or limit them to basics, like embeddings? Or not allow them at all?
      • Any tools to whitelist/blacklist?
  • Any general remarks?