Semantic/meaning representation parsing

- Old-school: With linguistically-precise hand-crafted grammars.
- Statistical: Learn grammar from (annotated) data.
- Neural: Learn to map with NN instead of grammar.
Compositional Neural Meaning Representation Parsing

Weiwei Sun

Natural Language and Information Processing Research Group
Department of Computer Science and Technology
University of Cambridge

December 5, 2020
How Can We Properly Represent Meanings?

Graph-based meaning representations have become popular

- Bi-lexical Semantic Dependency Graphs
How Can We Properly Represent Meanings?

Graph-based meaning representations have become popular

- Bi-lexical Semantic Dependency Graphs
- Abstract Meaning Representations (AMR)
How Can We Properly Represent Meanings?

Graph-based meaning representations have become popular

• Bi-lexical Semantic Dependency Graphs
• Abstract Meaning Representations (AMR)
• Dependency Minimal Recursion Semantics (DMRS)
How Can We Properly Represent Meanings?

Graph-based meaning representations have become popular

- Bi-lexical Semantic Dependency Graphs
- Abstract Meaning Representations (AMR)
- Dependency Minimal Recursion Semantics (DMRS)
- Elementary Dependency Structures (EDS)
(1) Every dog chases some cats.
(2) $\text{some}(y, \text{cat}(y), \text{every}(x, \text{dog}(x), \text{chase}(e, x, y)))$
(3) $\text{every}(x, \text{dog}(x), \text{some}(y, \text{cat}(y), \text{chase}(e, x, y)))$

```
  every(x)
     /
    /
  dog(x)  some(y)
        /
       /
  cat(y)  /
     /
  happy(e₂, e₁)  chase(e₁, x, y)
```
(1) Every dog chases some cats.
(2) \text{some}(y, \text{cat}(y), \text{every}(x, \text{dog}(x), \text{chase}(e, x, y)))
(3) \text{every}(x, \text{dog}(x), \text{some}(y, \text{cat}(y), \text{chase}(e, x, y)))
(1) Every dog chases some cats.
(2) \(\text{some}(y, \text{cat}(y), \text{every}(x, \text{dog}(x), \text{chase}(e, x, y)))\)
(3) \(\text{every}(x, \text{dog}(x), \text{some}(y, \text{cat}(y), \text{chase}(e, x, y)))\)
(1) Every dog chases some cats.

(2) \texttt{some}(y, \texttt{cat}(y), \texttt{every}(x, \texttt{dog}(x), \texttt{chase}(e, x, y)))

(3) \texttt{every}(x, \texttt{dog}(x), \texttt{some}(y, \texttt{cat}(y), \texttt{chase}(e, x, y)))

Logical Expression and Semantic Graph
(1) Every dog chases some cats.
(2) $\text{some}(y, \text{cat}(y), \text{every}(x, \text{dog}(x), \text{chase}(e, x, y)))$
(3) $\text{every}(x, \text{dog}(x), \text{some}(y, \text{cat}(y), \text{chase}(e, x, y)))$
(1) Every dog chases some cats.

(2) \texttt{some}(y, \texttt{cat}(y), \texttt{every}(x, \texttt{dog}(x), \texttt{chase}(e, x, y)))

(3) \texttt{every}(x, \texttt{dog}(x), \texttt{some}(y, \texttt{cat}(y), \texttt{chase}(e, x, y)))
(1) Every dog chases some cats.

(2) \text{some}(y, \text{cat}(y), \text{every}(x, \text{dog}(x), \text{chase}(e, x, y)))

(3) \text{every}(x, \text{dog}(x), \text{some}(y, \text{cat}(y), \text{chase}(e, x, y)))
(1) Every dog chases some cats.

(2) \[ \text{some}(y, \text{cat}(y), \text{every}(x, \text{dog}(x), \text{chase}(e, x, y))) \]

(3) \[ \text{every}(x, \text{dog}(x), \text{some}(y, \text{cat}(y), \text{chase}(e, x, y))) \]
String-to-Graph Parsing is Making Rapid Progress

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

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

• Inspired by the successful design of transition-based dependency parsers.
  

• Psycholinguistically motivated: Left-to-right, word-by-word

• Partially parsed results (parsing states) constrain parsing of subsequent words

• Greedy search to get a good parse.
Factorization-Based Approaches

- Explicitly modeling the target structure.
- A parser knows how to evaluate the goodnes of a candidate graph.
- A parser knows how to find the best graph from an extremely large set.
Modeling Syntactico-Semantic Composition

The Principle of Compositionality

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

B. Partee
Modeling Syntactico-Semantic Composition

The Principle of Compositionality

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

B. Partee
Modeling Syntactico-Semantic Composition

The Principle of Compositionality

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

B. Partee
Composition-Based Approaches

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

- A set of **nodes**.
- A set of **edges** connecting two nodes.
A hypergraph adds:

- **Hyperedges** connecting any number of nodes.
- A single node can be treated as an edge.
want to go to Saarbrücken
Hyperedge Replacement Grammar (Drewes et al., 1997)

want to go to Saarbrücken
Hyperedge Replacement Grammar (Drewes et al., 1997)

[Diagram of Hyperedge Replacement Grammar]

want to go to Saarbrücken
Hyperedge Replacement Grammar (Drewes et al., 1997)

VP

X

Z

Y

arg1
go
arg2
arg1
want
Saarbrücken
X

arg1
arg2
want
V

Y

Z

arg1
arg2
X

arg2
arg1
want
Saarbrücken
W

arg2
arg1
X

to go to Saarbrücken

want

VP

V

X

Y

Z

VP

12 of 32
Hyperedge Replacement Grammar (Drewes et al., 1997)

want to go to Saarbrücken
Hyperedge Replacement Grammar (Drewes et al., 1997)

Go to Saarbrücken

Want to go to Saarbrücken
Synchronous HRG

\[
S \rightarrow S \rightarrow NP \rightarrow VP \rightarrow NP \rightarrow V \rightarrow VP
\]

Linguistically-informed rules

\[
\begin{align*}
\gamma_1 & : S \Rightarrow \text{arg1} \\
\gamma_2 & : \text{arg1} \rightarrow \text{NP} \\
\gamma_3 & : \text{arg1} \rightarrow \text{VP} \\
\gamma_4 & : V \rightarrow \text{arg1} \rightarrow \text{arg2} \rightarrow \text{VP}
\end{align*}
\]

Control construction (Equi verbs) Generalized quantifier Predicate–argument structure
Derivation Structure is Tree; Derived Structure is Graph
Lexicalised Grammar

Some boys want to go

S

arg1
arg1
want

some bv boy

NP

NP

some bv boy

VP

good arg2 arg1 arg1

V

arg2 arg1 want

V

go arg1

∅ to go
Lexicalised Grammar

Some boys want
Lexicalised Grammar

More Complex Graph

every → BODY
 BV
 ARG0
 x
 RSTR

 ARG1

 dog

 ARG0

 e1

 ARG1

 happy

 e2

 ARG0

 chase

 ARG0

 ARG2

 y

 RSTR

 ARG0

 cat

 ARG0
Minimal Recursion Semantics (Copestake et al., 2005)
More Complex Derivation (Chen & Sun, 2020)

add types of nodes
Neural Graph Rewriting
Scoring a Derivation Tree

Semantic Graph Parsing

\[
\arg \max_{T \in \mathcal{T}(x)} \text{score}(T)
\]

Parsing Semantic Graphs

\[
\arg \max_{T \in \mathcal{T}(G)} \text{score}(T)
\]
Numbers of External Nodes

Count

Variable-reduced
Variable-in-situ

Numbers of External Nodes

1: 13,253
2: 3,486
3: 1,765
4: 1,945
5+: 526

Total: 32
Neural Graph Rewriting
Neural Graph Rewriting
Neural Graph Rewriting

[Diagram of neural graph rewriting with nodes and connections]
Neural Graph Rewriting
Neural Graph Rewriting
Vectorizing a Rule
Vectorizing a Rule
Vectorizing a Rule
Vectorizing a Rule
Recursive Hypergraph-state LSTM  (Chen & Sun, 2020)
Recursive Hypergraph-state LSTM  (Chen & Sun, 2020)
Recursive Hypergraph-state LSTM  (Chen & Sun, 2020)
Recursive Hypergraph-state LSTM  (Chen & Sun, 2020)
String-to-Graph Parsing (Chen et al., 2018)

Syntactic parsing + Semantic interpretation ⇒ Meaning representation
String-to-Graph Parsing (Chen et al., 2018)

Syntactic parsing + Semantic interpretation ⇒ Meaning representation

### Syntactic parsing
- **Word representation:** LSTM/Transformer
- **Phrase representation:** LSTM-minus/Transformer

![Diagram of a parse tree and representations](image)

```
S
  NP
    D: Some
    N: boys
  VP
    V: want
    VP
      V: to
      V: go
```

```
want to go
LSTM
LSTM
LSTM
LSTM
```

```
boys
want
to
```

```
go
```
String-to-Graph Parsing (Chen et al., 2018)

Syntactic parsing + Semantic interpretation ⇒ Meaning representation
String-to-Graph Parsing

Chen et al., 2018

Syntactic parsing + Semantic interpretation \(\Rightarrow\) Meaning representation

Semantic interpretation

- Rule embedding
- Subgraph embedding
- Beam search decoder
String-to-Graph Parsing  
(Chen et al., 2018)

Syntactic parsing + Semantic interpretation ⇒ Meaning representation

Semantic interpretation

• Rule embedding
• Subgraph embedding
• Beam search decoder
EMPIRICAL EVALUATION
The boy wants to go.
Evaluation for Parsing to EDS/DMRS

\[ \text{0 } \text{The 1 } \text{boy 2 wants 3 to 4 go 5 .} \]

_\text{the}_q(0,1) _\text{want}_v\text{to}(2,3)

_\text{boy}_n\text{-1}(1,2) _\text{go}_v\text{-1}(4,5)
Evaluation for Parsing to EDS/DMRS

The boy wants to go.

_boy_n_1(1,2)  _want_v_to(2,3)

_go_v_1(4,5)  _the_q(0,1)

\[ V_{gold} = \{(0, 1), _\text{the_q}\}, \ldots\}, |V_{gold}| = 4 \]
Evaluation for Parsing to EDS/DMRS

The boy wants to go.

\[ \text{the_q}(0,1) \quad \text{want_v_to}(2,3) \]

\[ \text{boy_n_1}(1,2) \quad \text{go_v_1}(4,5) \]

\( V_{\text{gold}} = \{ (\langle 0, 1 \rangle, \text{the_q}), \cdots \} \), \( |V_{\text{gold}}| = 4 \)
Evaluation for Parsing to EDS/DMRS

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

\[
\begin{array}{c}
\text{the}_q(0,1) \\
\text{boy}_n_1(1,2) \\
\text{want}_v_{to}(2,3) \\
\text{go}_v_1(4,5)
\end{array}
\]

\[
V_{\text{gold}} = \{(0,1), \text{the}_q\}, \ldots \}, |V_{\text{gold}}| = 4
\]

\[
E_{\text{gold}} = \{(0,1), BV, (1,2)\}, \ldots \}, |E_{\text{gold}}| = 4
\]
Evaluation for Parsing to EDS/DMRS

The boy wants to go.

\[
\begin{align*}
V_{\text{gold}} &= \{(0,1, \_\text{the}_q), \ldots \}, \quad |V_{\text{gold}}| = 4 \\
E_{\text{gold}} &= \{(0,1, \_\text{boy}_n_1, 1, 2), \ldots \}, \quad |E_{\text{gold}}| = 4
\end{align*}
\]

Dridan & Oepen (2011)
Evaluation for Parsing to EDS/DMRS

\[ The 1 \text{ boy } 2 \text{ wants } 3 \text{ to } 4 \text{ go } 5. \]

\[ _\text{the}_q(0,1) \quad _\text{want}_v_\text{to}(2,3) \quad _\text{the}_q(0,1) \quad _\text{want}_v_\text{to}(2,3) \]

\[ _\text{boy}_n_1(1,2) \quad _\text{go}_v_1(4,5) \quad _\text{boy}_n_1(1,2) \quad _\text{go}_n_1(4,5) \]

\[ V_{\text{gold}} = \left\{ (\langle 0,1 \rangle, _\text{the}_q), \cdots \right\}, \quad |V_{\text{gold}}| = 4 \]
\[ E_{\text{gold}} = \left\{ (\langle 0,1 \rangle, _\text{bv}, \langle 1,2 \rangle), \cdots \right\}, \quad |E_{\text{gold}}| = 4 \]

\[ V_{\text{sys}} = \left\{ (\langle 0,1 \rangle, _\text{the}_q), \cdots \right\}, \quad |V_{\text{sys}}| = 4 \]

Dridan & Oepen (2011)
Evaluation for Parsing to EDS/DMRS

\[ \text{The boy wants to go}. \]

\[ \text{V}_{\text{gold}} = \{(\langle 0, 1 \rangle, \text{_the_q}), \cdots \}, |\text{V}_{\text{gold}}| = 4 \]
\[ \text{E}_{\text{gold}} = \{(\langle 0, 1 \rangle, \text{BV}, \langle 1, 2 \rangle), \cdots \}, |\text{E}_{\text{gold}}| = 4 \]

\[ \text{V}_{\text{sys}} = \{(\langle 0, 1 \rangle, \text{_the_q}), \cdots \}, |\text{V}_{\text{sys}}| = 4 \]

\[ \text{E}_{\text{match}} = \{(\langle 1, 2 \rangle, \text{ARG1}), \cdots \}, |\text{E}_{\text{match}}| = 2 \]

\[ \text{EDM}_{\text{na}} = 2^* \]

Dridan & Oepen (2011)
Evaluation for Parsing to EDS/DMRS

_The_ 1 _boy_ 2 _wants_ 3 _to_ 4 _go_ 5 . 6

\[\text{V}_{\text{gold}} = \{(0, 1), \_\text{the\_q}\}, \ldots\}, \ |\text{V}_{\text{gold}}| = 4 \]
\[\text{E}_{\text{gold}} = \{(0, 1), \_\text{the\_q}\}, \ldots\}, \ |\text{E}_{\text{gold}}| = 4 \]

\[\text{V}_{\text{sys}} = \{(0, 1), \_\text{the\_q}\}, \ldots\}, \ |\text{V}_{\text{sys}}| = 4 \]
\[\text{E}_{\text{sys}} = \{(1, 2), \_\text{go\_v}\}, \ldots\}, \ |\text{E}_{\text{sys}}| = 3 \]
Evaluation for Parsing to EDS/DMRS

\[0 \text{ The 1 boy 2 wants 3 to 4 go 5 .} 6\]

\[\text{the}_q(0, 1)\]
\[\text{want}_v\text{to}(2, 3)\]
\[\text{boy}_n(1, 2)\]
\[\text{go}_v(4, 5)\]

\[V_{\text{gold}} = \{(\langle 0, 1 \rangle, \text{the}_q), \cdots \}, |V_{\text{gold}}| = 4\]
\[E_{\text{gold}} = \{(\langle 0, 1 \rangle, \text{bv}, \langle 1, 2 \rangle), \cdots \}, |E_{\text{gold}}| = 4\]

\[V_{\text{sys}} = \{(\langle 0, 1 \rangle, \text{the}_q), \cdots \}, |V_{\text{sys}}| = 4\]
\[E_{\text{sys}} = \{(\langle 1, 2 \rangle, \text{bv}, \langle 2, 1 \rangle), \cdots \}, |E_{\text{sys}}| = 3\]

\[V_{\text{match}} = V_{\text{gold}} \cap V_{\text{sys}} = \{(\langle 1, 2 \rangle, \text{boy}_n, 1), \cdots \} |V_{\text{match}}| = 3\]
\[E_{\text{match}} = E_{\text{gold}} \cap E_{\text{sys}} = \{(\langle 2, 3 \rangle, \text{ARG1}, \langle 1, 2 \rangle), \cdots \} |E_{\text{match}}| = 2\]

Dridan & Oepen (2011)
Evaluation for Parsing to EDS/DMRS

\[ 0 \text{ The 1 boy 2 wants 3 to 4 go 5 .} \]

\[
\begin{align*}
V_{\text{gold}} & = \{(\langle 0, 1 \rangle, _{\text{the_q}}), \cdots \}, \quad |V_{\text{gold}}| = 4 \\
E_{\text{gold}} & = \{(\langle 0, 1 \rangle, _{\text{bv}}, \langle 1, 2 \rangle), \cdots \}, \quad |E_{\text{gold}}| = 4
\end{align*}
\]

\[
\begin{align*}
V_{\text{sys}} & = \{(\langle 0, 1 \rangle, _{\text{the_q}}), \cdots \}, \quad |V_{\text{sys}}| = 4 \\
E_{\text{sys}} & = \{(\langle 1, 2 \rangle, _{\text{bv}}, \langle 2, 1 \rangle), \cdots \}, \quad |E_{\text{sys}}| = 3
\end{align*}
\]

\[
\begin{align*}
V_{\text{match}} & = V_{\text{gold}} \cap V_{\text{sys}} = \{(\langle 1, 2 \rangle, _{\text{boy_n_1}}), \cdots \} \quad |V_{\text{match}}| = 3 \\
E_{\text{match}} & = E_{\text{gold}} \cap E_{\text{sys}} = \{(\langle 2, 3 \rangle, _{\text{ARG1}}, \langle 1, 2 \rangle), \cdots \} \quad |E_{\text{match}}| = 2
\end{align*}
\]

\[
\begin{align*}
\text{EDM}_n & = \frac{2^*|V_{\text{match}}|}{|V_{\text{gold}}|+|V_{\text{sys}}|} = 0.86 \\
\text{EDM}_a & = \frac{2^*|E_{\text{match}}|}{|E_{\text{gold}}|+|E_{\text{sys}}|} = 0.57 \\
\text{EDM}_{\text{na}} & = \frac{2^*(|V_{\text{match}}|+|E_{\text{match}}|)}{|V_{\text{gold}}|+|V_{\text{sys}}|+|E_{\text{gold}}|+|E_{\text{sys}}|} = 0.67
\end{align*}
\]

Dridan & Oepen (2011)
Evaluation for Parsing to AMR

EDS/DMRS free node–sub-string correspondences
AMR no sub-string correspondences annotated

Diagram:

- **_boy_n_1**
  - ARG0
  - ARG0

- **_want_v_1**
  - ARG1
  - **_go_v_1**

A
B
C
Evaluation for Parsing to AMR

EDS/DMRS free node–sub-string correspondences
AMR no sub-string correspondences annotated

ARG0
ARG1
ARG0
ARG1
ARG0
ARG1
ARG0
Evaluation for Parsing to AMR

EDS/DMRS  free node–sub-string correspondences
AMR       no sub-string correspondences annotated

Assume an alignment:  \( A \leftrightarrow X \)  &  \( B \leftrightarrow Y \)  &  \( C \leftrightarrow Z \)
Evaluation for Parsing to AMR

EDS/DMRS  free node–sub-string correspondences
AMR        no sub-string correspondences annotated

Assume an alignment:  \( A \leftrightarrow X \) &  \( B \leftrightarrow Y \) &  \( C \leftrightarrow Z \)
Evaluation for Parsing to AMR

EDS/DMRS free node–sub-string correspondences
AMR no sub-string correspondences annotated

\[
\begin{align*}
_{\text{boy}_n_1} & \rightarrow \text{ARG0} \rightarrow \_\text{want}_v_1 \rightarrow \text{ARG1} \rightarrow \_\text{go}_v_1 \rightarrow \text{ARG0} \\
\_\text{go}_v_1 & \rightarrow \text{ARG0} \rightarrow \_\text{boy}_n_1 \rightarrow \text{ARG0} \\
\end{align*}
\]

\[
\begin{align*}
EDM_n & = \frac{2 \times 2}{3+3} = 0.67 \\
EDM_a & = \frac{2 \times 1}{3+2} = 0.40 \\
EDM_{na} & = \frac{2 \times 3}{6+5} = 0.55
\end{align*}
\]
Evaluation for Parsing to AMR

EDS/DMRS  free node–sub-string correspondences
AMR      no sub-string correspondences annotated

There are many such alignments: $A \leftrightarrow X \& B \leftrightarrow Z \& C \leftrightarrow Y$
Evaluation for Parsing to AMR

EDS/DMRS free node–sub-string correspondences
AMR no sub-string correspondences annotated

There are many such alignments: \( A \leftrightarrow X \) & \( B \leftrightarrow Z \) & \( C \leftrightarrow Y \)
Evaluation for Parsing to AMR

EDS/DMRS free node–sub-string correspondences
AMR no sub-string correspondences annotated

\[
\begin{align*}
\text{EDM}_n & \quad \text{EDM}_a & \quad \text{EDM}_{na} \\
\frac{2 \times 1}{3+3} = 0.33 & \quad \frac{2 \times 1}{3+2} = 0.40 & \quad \frac{2 \times 2}{6+5} = 0.36
\end{align*}
\]
Evaluation for Parsing to AMR

EDS/DMRS free node–sub-string correspondences
AMR no sub-string correspondences annotated

\[
S_{\text{MATCH}}(G_g, G_s) = \max_{a \in \mathcal{A}(G_g, G_s)} \text{EDM}_{\text{na}}(a)
\]

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

Cai & Knight (2013)
Structural Validation

```
<table>
<thead>
<tr>
<th>dog</th>
<th>every(x)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

\[ \text{every}(x) \]

\[ \text{dog}(x) \quad ??? \]

\text{semantic graph} \Rightarrow \text{underspecification}
```
Structural Validation as a Search Problem

\{h_1 : \text{every}'(x, h_6, h_7), h_2 : \text{dog}'(x),
  h_3 : \text{chase}'(x, y), h_4 : \text{some}'(y, h_8, h_9),
  h_5 : \text{cat}'(y),
  h_6 = q h_2, h_8 = q h_5, h_0 = q h_3\}
Structural Validation as a Search Problem

\{h_1 : \text{every}'(x, h_6, h_7), h_2 : \text{dog}'(x), h_3 : \text{chase}'(x, y), h_4 : \text{some}'(y, h_8, h_9), h_5 : \text{cat}'(y), h_6 =_q h_2, h_8 =_q h_5, h_0 =_q h_3\}\n
NP-hard; search in a smart way
Two state-of-the-art systems from CoNLL 2019 shared task
## Magic Numbers

### Accuracy

<table>
<thead>
<tr>
<th></th>
<th>DM id</th>
<th>PAS id</th>
<th>PSD id</th>
<th>EDS Smatch</th>
<th>AMR Smatch</th>
<th>PAS ood</th>
<th>PSD ood</th>
<th>EDS EDM</th>
<th>AMR Smatch</th>
</tr>
</thead>
<tbody>
<tr>
<td>Groschwitz et al. (2018)</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Lyu &amp; Titov (2018)</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Zhang et al. (2019)</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>H. Peng et al. (2017)</td>
<td>89.4</td>
<td>84.5</td>
<td>92.2</td>
<td>88.3</td>
<td>77.6</td>
<td>75.3</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>+Multitask learning</td>
<td>90.4</td>
<td>85.3</td>
<td>92.7</td>
<td>89.0</td>
<td>78.5</td>
<td>76.4</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Dozat &amp; Manning (2018)</td>
<td>93.7</td>
<td>88.9</td>
<td>94.0</td>
<td>90.8</td>
<td>81.0</td>
<td>79.4</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Buys &amp; Blunsom (2017)</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>85.5</td>
<td>85.9</td>
<td>60.1</td>
</tr>
<tr>
<td>Chen et al. (2018)</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>90.9</td>
<td>90.4</td>
<td>-</td>
</tr>
<tr>
<td>Lindemann et al. (2019)</td>
<td>93.9</td>
<td>90.3</td>
<td>94.5</td>
<td>92.5</td>
<td>82.0</td>
<td>81.5</td>
<td>90.1</td>
<td>84.9</td>
<td>74.3</td>
</tr>
<tr>
<td>+Multitask learning</td>
<td>94.1</td>
<td>90.5</td>
<td>94.7</td>
<td>92.8</td>
<td>82.1</td>
<td>81.6</td>
<td>90.4</td>
<td>85.2</td>
<td>74.5</td>
</tr>
</tbody>
</table>

from Lindemann et al. (2019)

<table>
<thead>
<tr>
<th></th>
<th>DM id</th>
<th>PAS id</th>
<th>PSD id</th>
<th>EDS Smatch</th>
<th>AMR Smatch</th>
<th>PAS ood</th>
<th>PSD ood</th>
<th>EDS EDM</th>
<th>AMR Smatch</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chen &amp; Sun (2020)</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>93.1</td>
<td>92.4</td>
</tr>
</tbody>
</table>

### Structural Validation

For DMRS parsing, 92.17% graphs are logically coherent.
What is the meaning of life?

_life_v_1

[Diagram]
What is the meaning of life?

- Modeling syntactico-semantic composition
  - Graph-structured syntax–semantics interface
- Neural-Symbolic Integration
  - Neural graph rewriting
    recursive Graph Neural Networks
- Bi-directional parsing $\Rightarrow$ graph parsing
- Flexible to a wide range of semantic graphs
  - Variable-reduced conceptual graphs
  - Variable-in-situ logico-semantic graphs
\[ \mathcal{R} \rightarrow \text{Natural Language Sentence} \]

**Semantic Graph Parsing**

\[
\arg \max_{T \in \mathcal{T}(x)} \text{SCORE}(T)
\]

**Parsing Semantic Graphs**

\[
\arg \max_{T \in \mathcal{T}(G)} \text{SCORE}(T)
\]

Ye & Sun (2020)
\[ \mathcal{R} \rightarrow \text{Natural Language Sentence} \]

**Semantic Graph Parsing**

\[
\arg \max_{T \in \mathcal{T}(x)} \text{SCORE}(T)
\]

**Parsing Semantic Graphs**

\[
\arg \max_{T \in \mathcal{T}(G)} \text{SCORE}(T)
\]

Ye & Sun (2020)
Natural Language Sentence

Semantic Graph Parsing
\[
\arg \max_{T \in \mathcal{T}(x)} \text{score}(T)
\]

Parsing Semantic Graphs
\[
\arg \max_{T \in \mathcal{T}(G)} \text{score}(T)
\]
Thank You for Your Attention


References III


References IV


References V


References VI


