Graph-Based Meaning Representations: Design and Processing

GYN

Graph-Based Meaning Bank

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

- **Abstract Meaning Representation(AMR)**: encodes sentence-level semantics, such as **predicate-argument information**, **reentrancies**, **named entities**, **negation** and **modality**, into a **rooted, directed**, and usually acyclic graph with **node and edge labels**. AMR graphs abstract away from syntactic realizations, i.e., there is no explicit correspondence between elements of the graph and the surface utterance.
- **Semantic Dependency Parsing(SDP)**: Their annotations have been converted into bi-lexical dependencies, forming directed graphs whose **nodes injectively correspond to surface lexical units**, and edges represent semantic relations between nodes (DM, PAS, PSD)
- **Universal Conceptual Cognitive Annotation(UCCA)**: targets a level of semantic granularity that abstracts away from syntactic paraphrases in a typologically motivated, cross-linguistic fashion. Sentence representations in UCCA are directed acyclic graphs (DAG), where terminal nodes correspond to **surface lexical tokens**, and non-terminal nodes to semantic units that participate in super-ordinate relations. Edges are labeled, indicating the role of a child in the relation the parent represents.

AMR

- AMR aims to abstract away from syntactic idiosyncrasies. We attempt to assign the same AMR to sentences that have the same basic meaning.
- AMR makes extensive use of PropBank framesets (Kingsbury and Palmer, 2002; Palmer et al., 2005). For example, we represent a phrase like “bond investor” using the frame “invest-01”, even though no verbs appear in the phrase.
- AMR is agnostic about how we might want to derive meanings from strings, or vice-versa. This makes sembanking very fast, and it allows researchers to explore their own ideas about how strings are related to meanings.

LOGIC format:

$$\exists\, w, b, g:$$
$$\text{instance}(w, \text{want-01}) \wedge \text{instance}(g, \text{go-01}) \wedge$$
$$\text{instance}(b, \text{boy}) \wedge \text{arg0}(w, b) \wedge$$
$$\text{arg1}(w, g) \wedge \text{arg0}(g, b)$$

AMR format (based on PENMAN):

```
(w / want-01
  :arg0 (b / boy)
  :arg1 (g / go-01
          :arg0 b))
```

GRAPH format:

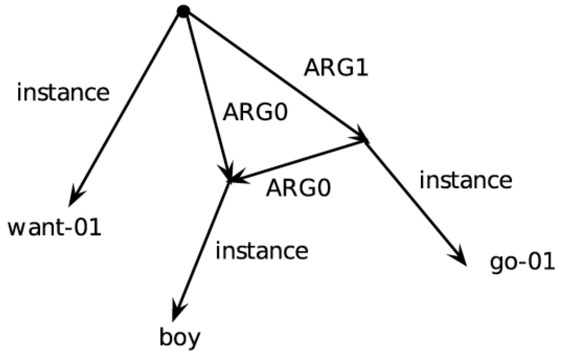


Figure 1: Equivalent formats for representating the meaning of “The boy wants to go”.

Using AMR for Abstract Meaning Representation

Toward Abstractive Summarization Using Semantic Representations

Fei Liu Jeffrey Flanigan Sam Thomson Norman Sadeh Noah A. Smith
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213, USA
{feiliu, jflanigan, sthomson, sadeh, nasmith}@cs.cmu.edu

In this framework,

1. parsing the input sentences to individual AMR graphs (JAMR (Flanigan et al., 2014), 63% F-score).
2. **combining and transforming those graphs into a single summary AMR graph**
3. generating text from the summary graph

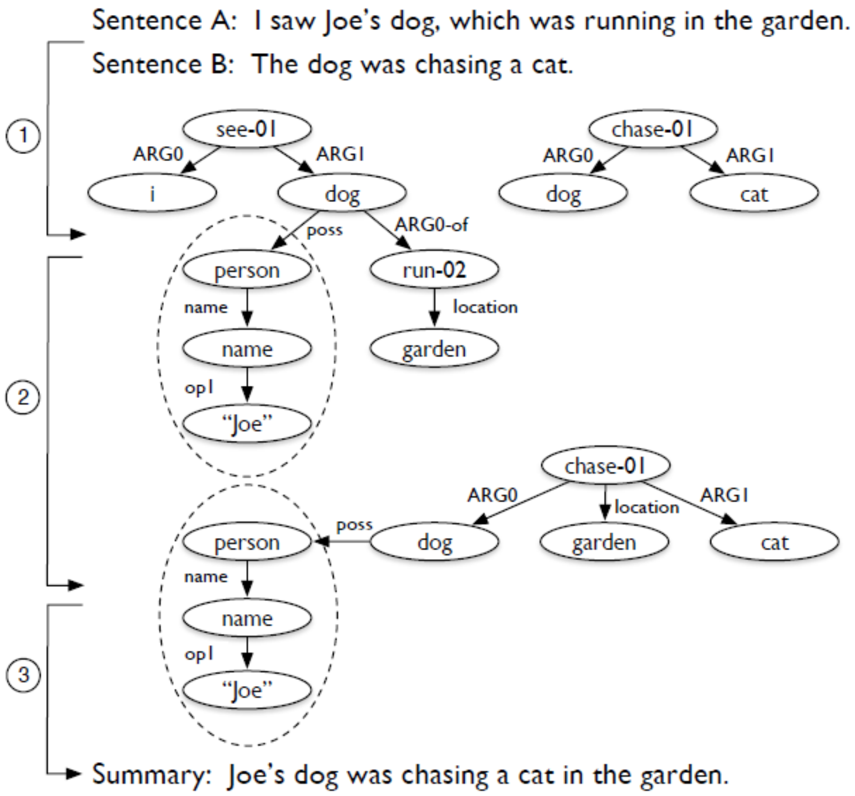
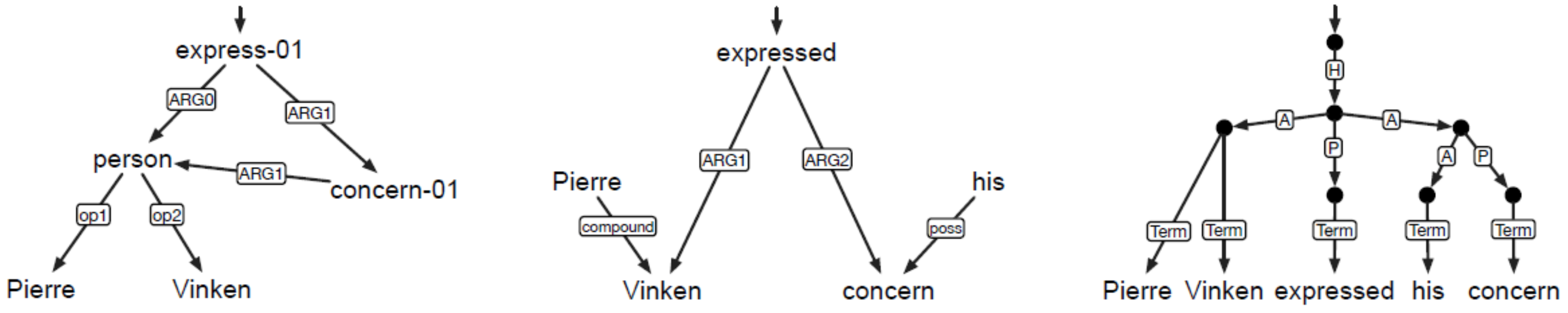


Figure 1: A toy example. Sentences are parsed into individual AMR graphs in step 1; step 2 conducts graph transformation that produces a single summary AMR graph; text is generated from the summary graph in step 3.

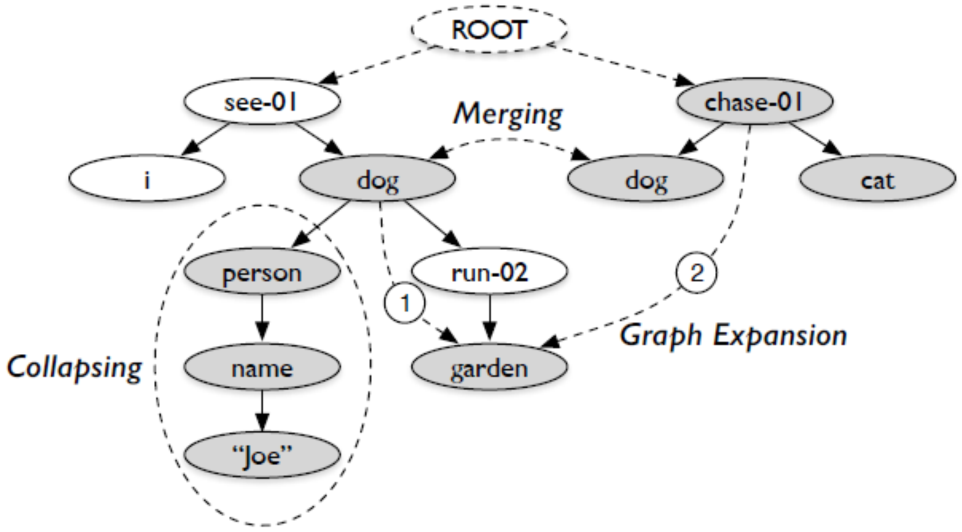
Graph-Based Meaning Bank

- **“Pierre Vinken expressed his concern”.**



(a) AMR (b) DM (c) UCCA

Using AMR for Abstract Meaning Representation



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

Figure 3: A source graph formed from two sentence AMR graphs. Concept collapsing, merging, and graph expansion are demonstrated. Edges are unlabeled. A "ROOT" node is added to ensure connectivity. (1) and (2) are among edges added through the optional expansion step, corresponding to sentence- and document-level expansion, respectively. Concept nodes included in the summary graph are shaded.

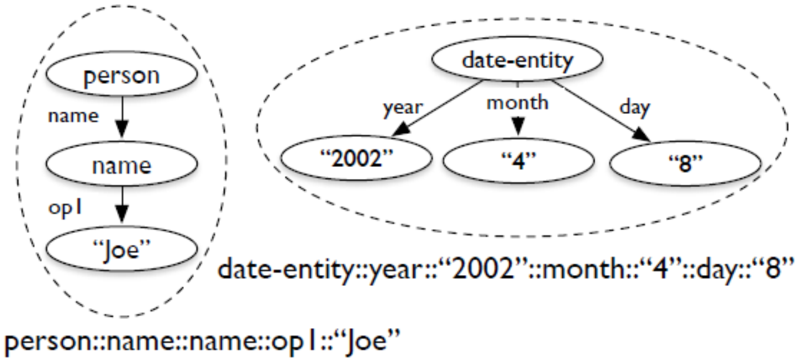
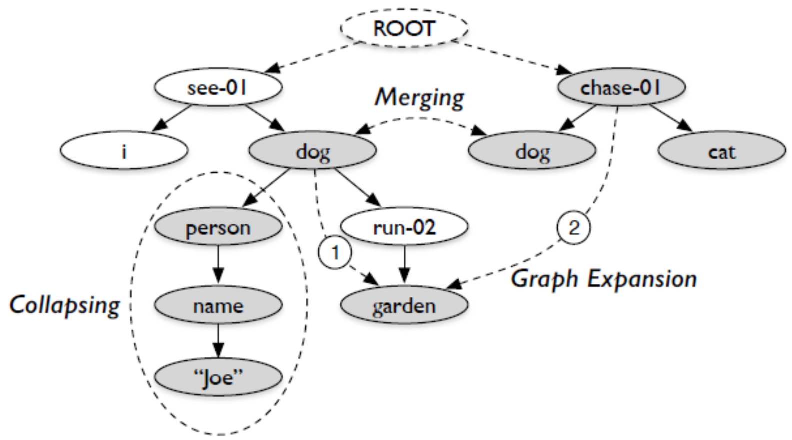


Figure 2: Graph fragments are collapsed into a single concept and assigned a new concept label.

1. Source Graph Construction
   1. "source graph" is a single graph constructed using the individual sentences' AMR graphs by merging identical concepts.
   2. **Concept collapsing, merging, and graph expansion [sentence-level]**
   3. * won't recognize coreferent concepts like "Barack Obama" = "Obama" and "say-01" = "report-01,"

Using AMR for Abstract Meaning Representation



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

Sentence B: The dog was chasing a cat.

Figure 3: A source graph formed from two sentence AMR graphs. Concept collapsing, merging, and graph expansion are demonstrated. Edges are unlabeled. A "ROOT" node is added to ensure connectivity. (1) and (2) are among edges added through the optional expansion step, corresponding to sentence- and document-level expansion, respectively. Concept nodes included in the summary graph are shaded.

1. Subgraph Prediction
    1. seek to maximize a score that factorizes over graph nodes and edges that are included in the summary graph. (integer linear programming)

$$score(V', E'; \boldsymbol{\theta}, \boldsymbol{\psi}) = \sum_{v \in V'} \boldsymbol{\theta}^\top \mathbf{f}(v) + \sum_{e \in E'} \boldsymbol{\psi}^\top \mathbf{g}(e) \quad (1)$$

$$\sum_{i=1}^{N} v_i \underbrace{\boldsymbol{\theta}^\top \mathbf{f}(i)}_{\text{node score}} + \sum_{(i,j) \in E} e_{i,j} \underbrace{\boldsymbol{\psi}^\top \mathbf{g}(i,j)}_{\text{edge score}} \quad (2)$$

Constraints are required to ensure that the selected nodes and edges form a valid graph. In particular, if an edge (i, j) is selected ($e_{i,j}$ takes value of 1), then both its endpoints i, j must be included:

$$v_i - e_{i,j} \geq 0, \quad v_j - e_{i,j} \geq 0, \quad \forall i, j \leq N \quad (3)$$

Connectivity is enforced using a set of single-commodity flow variables $f_{i,j}$, each taking a non-negative integral value, representing the flow from node i to j. The root node sends out up to N units of flow, one to reach each included node (Equation 4). Each included node consumes one unit of flow, reflected as the difference between incoming and outgoing flow (Equation 5). Flow may only be sent over an edge if the edge is included (Equation 6).

$$\sum_i f_{0,i} - \sum_i v_i = 0, \quad (4)$$

$$\sum_i f_{i,j} - \sum_k f_{j,k} - v_j = 0, \quad \forall j \leq N, \quad (5)$$

$$N \cdot e_{i,j} - f_{i,j} \geq 0, \quad \forall i, j \leq N. \quad (6)$$

Using AMR for Abstract Meaning Representation

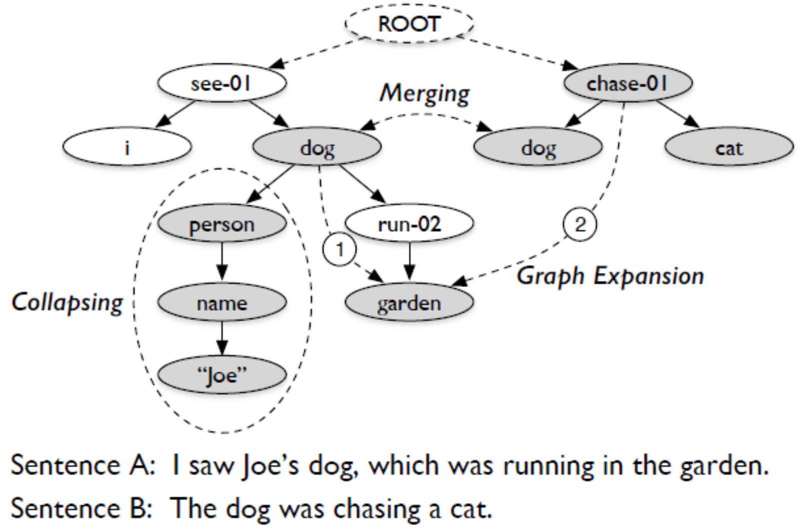


Figure 3: A source graph formed from two sentence AMR graphs. Concept collapsing, merging, and graph expansion are demonstrated. Edges are unlabeled. A "ROOT" node is added to ensure connectivity. (1) and (2) are among edges added through the optional expansion step, corresponding to sentence- and document-level expansion, respectively. Concept nodes included in the summary graph are shaded.

Generation:

- Given a predicted subgraph, a system summary is created by finding the most frequently aligned word span for each concept node).
    - In addition to predicting AMR graphs for each sentence, JAMR provides alignments between spans of words in the source sentence and fragments of its predicted graph. For example, a graph fragment headed by "date-entity" could be aligned to the tokens "April 8, 2002." We use these alignments in our simple text generation module.
- The words in the resulting spans are generated in no particular order.
- While this is not a natural language summary, it is suitable for unigram-based summarization evaluation methods like ROUGE-1.

Using AMR for Abstract Meaning Representation

Guided Neural Language Generation for Abstractive Summarization using Abstract Meaning Representation

Hardy
The University of Sheffield
hhardy2@sheffield.ac.uk

Andreas Vlachos
The University of Sheffield
a.vlachos@sheffield.ac.uk

Objective:

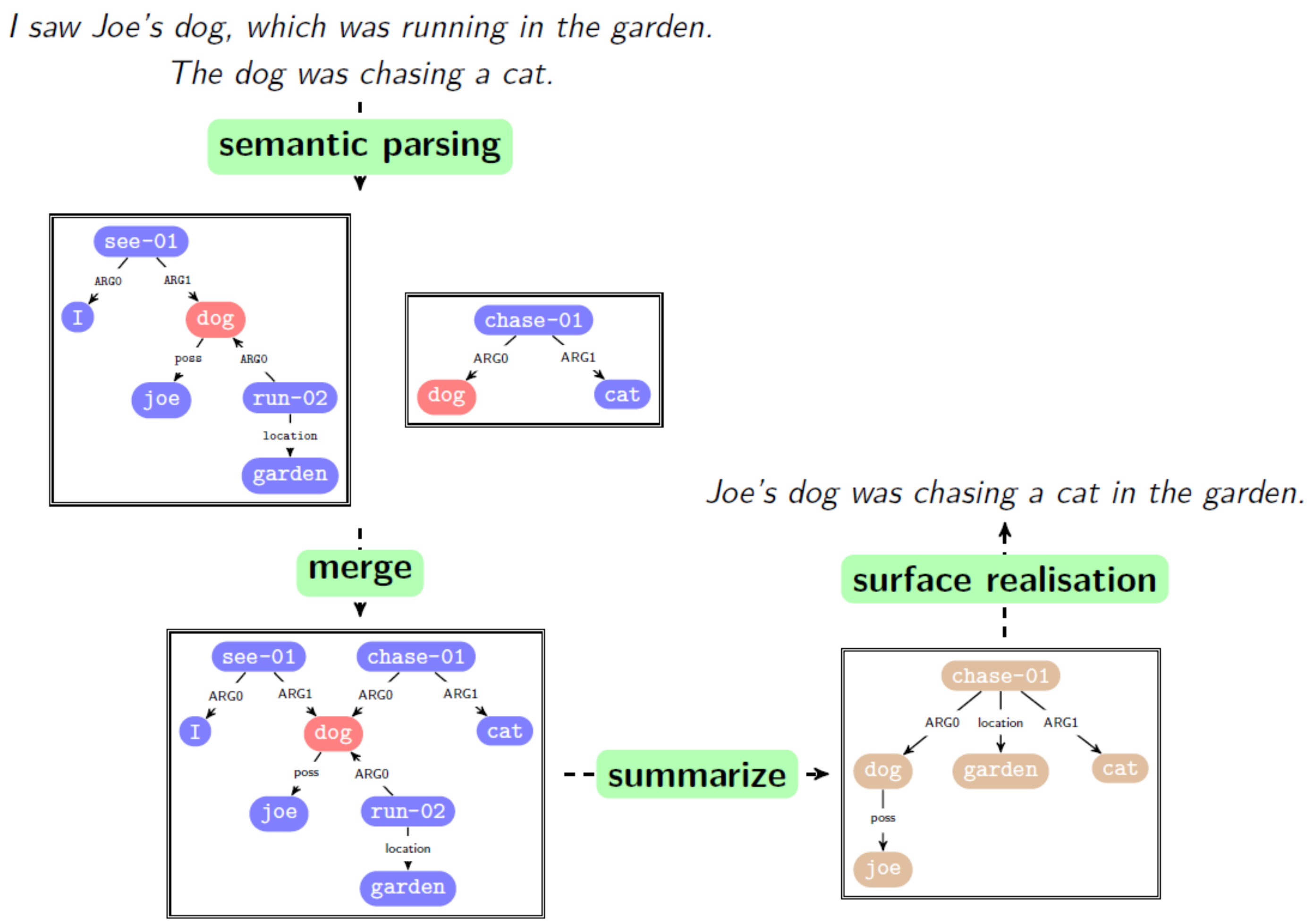
1. Retrieve the information missing from AMR but needed for NLG.
2. Improve the quality of the summary.

- Unguided NLG from AMR
  - Input AMR and generate target words
- Guided NLG from AMR
  - **side information**: we prune the source document to a set of k sentences which have the highest similarity with the summary AMR graph.
  - <用LCS选出topk的句子作为生成的补丁>

$$s(y_j|y_{<j}, z) = log\, a + \psi * log(\frac{b}{a} + 1) \quad (8)$$

where ψ is a hyper-parameter determining the influence of the side information on the decoding process, a is $P_{s2s}(y_j|y_{<j}, z)$ and b is $P_{side}(y_j|y_{j-3}^{j-1})$. $s(y_j|y_{<j}, z)$ is used during beam search replacing $P_{s2s}(y_j|y_{<j}, z)$ for all words that occur in the side information. The intuition behind Eq. 8 is that we are rewarding word y_j when it appears in similar context in the side information, i.e. the source document being summarized.

Using AMR for Abstract Meaning Representation



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

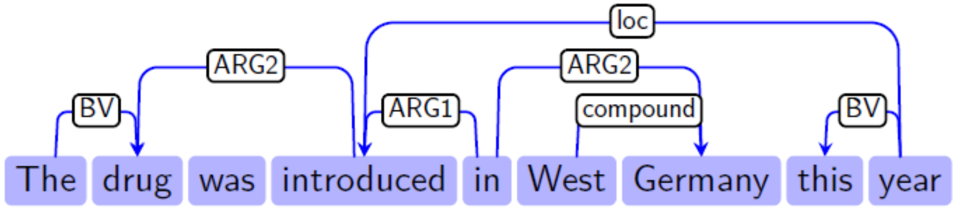
Parsing to Flavor (0) Graphs

Parsing to Flavor (0) Graphs

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

Parsing to Flavor (0) graphs

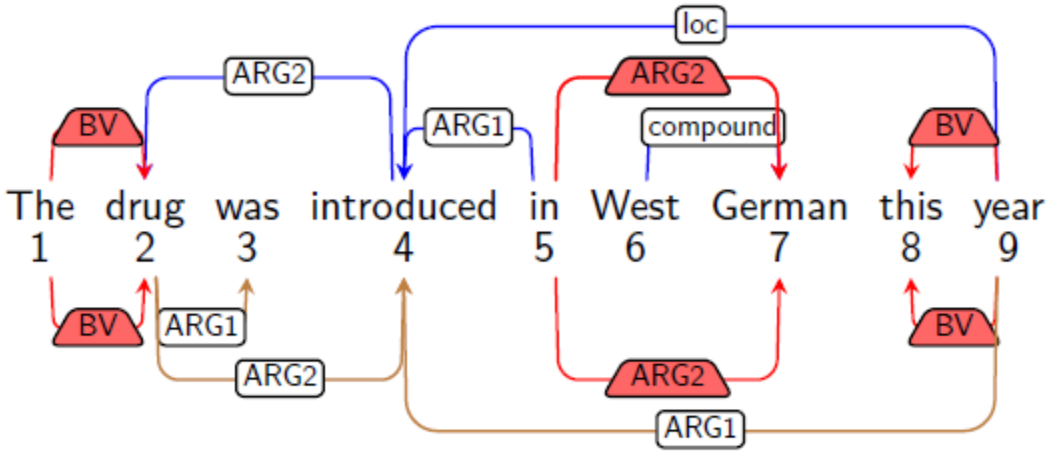
- Nodes = tokens
- The goal is to predict labeled edges



Semantic Parsing is Making Rapid Progress

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

Evaluation for Parsing to Flavor (0) Graphs



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

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

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

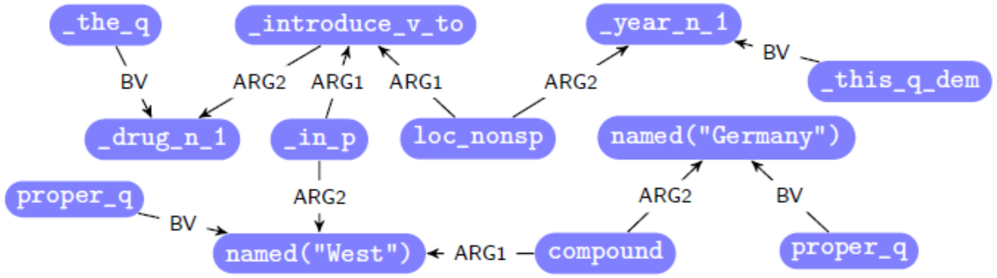
Precision	Recall	F-score
$\frac{\lvert E_{\text{match}}\rvert}{\lvert E_{\text{system}}\rvert} = 0.43$	$\frac{\lvert E_{\text{match}}\rvert}{\lvert E_{\text{gold}}\rvert} = 0.5$	$\frac{2 * \lvert E_{\text{match}}\rvert}{\lvert E_{\text{gold}}\rvert + \lvert E_{\text{system}}\rvert} = 0.46$

Parsing to Flavor (1)(2) Graphs

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

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

- We need to predict labeled nodes and labeled edges

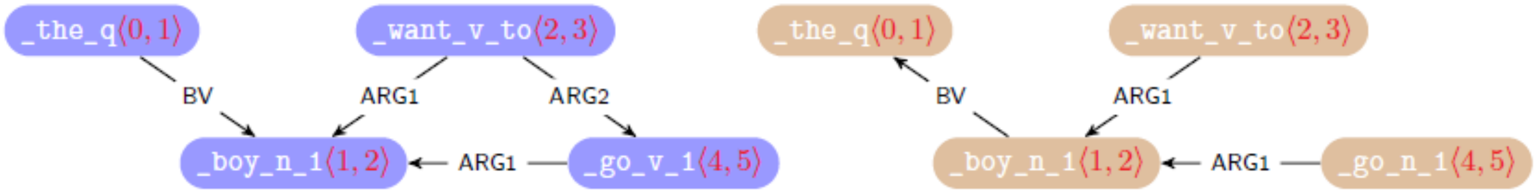


Semantic Parsing is Making Rapid Progress

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

Evaluation for Parsing to Flavor (1) Graphs[13]

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



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

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

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

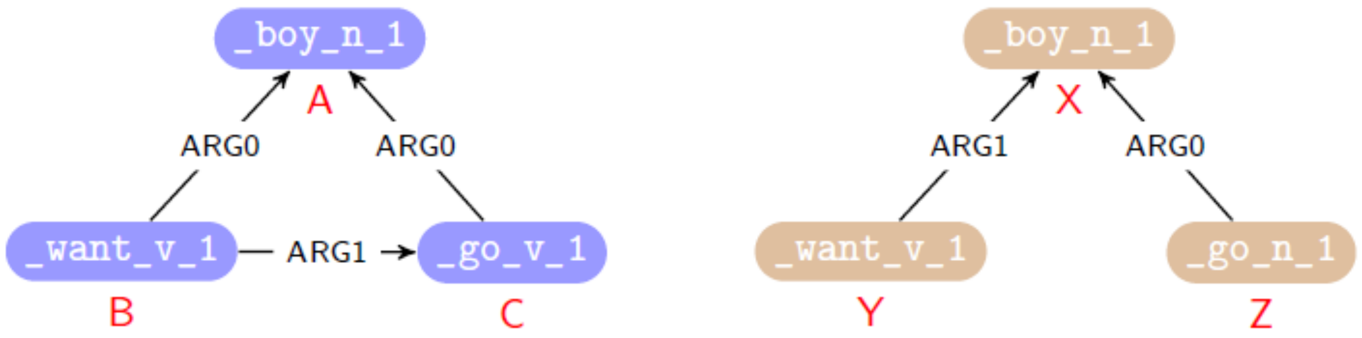
$$E_{\text{match}} = E_{\text{gold}} \cap E_{\text{sys}} = \{(\langle 2,3\rangle, \text{ARG1}, \langle 1,2\rangle), \cdots\} \ |E_{\text{match}}| = 2$$

EDM_n	EDM_a	EDM_na
$\frac{2*\lvert V_{\text{match}}\rvert}{\lvert V_{\text{gold}}\rvert+\lvert V_{\text{sys}}\rvert} = 0.86$	$\frac{2*\lvert E_{\text{match}}\rvert}{\lvert E_{\text{gold}}\rvert+\lvert E_{\text{sys}}\rvert} = 0.57$	$\frac{2*(\lvert V_{\text{match}}\rvert+\lvert E_{\text{match}}\rvert)}{\lvert V_{\text{gold}}\rvert+\lvert V_{\text{sys}}\rvert+\lvert E_{\text{gold}}\rvert+\lvert E_{\text{sys}}\rvert} = 0.67$

Dridan & Oepen (2011)

Evaluation for Parsing to Flavor (2) Graphs[14]

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



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

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

Cai & Knight (2013)

Sub-Tasks

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

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