Natural Language Processing

Lecture 11: Semantic Parsing - Abstract Meaning Representation (AMR)

11/30/2020

COMS W4705
Yassine Benajiba

Logical Forms

- Logical form satisfies many goals for meaning representations (unambiguous, canonical form, supports inference, expressiveness)
- But difficult to annotate on a large scale.

We skipped this, so let’s briefly talk about it

Abstract Meaning Representation (AMR)

(Banarescu et al., 2013)

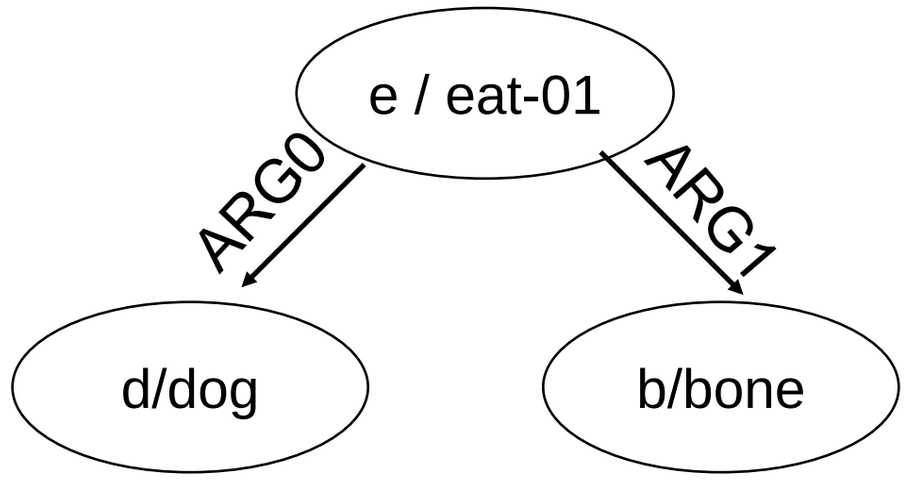
- Uses a single, simple data structure (feature structures / directed graphs) to represent many aspects of meaning.
  - Focus on "who does what to whom" but leave out details (tense, quantifiers, etc.)
  - This level of abstraction facilitates **consistent, large-scale human annotation.**
    Goal: build a giant "semantics bank" (comparable to treebanks for syntax).

(amr.isi.edu)

AMR Example

The dog is eating a bone.

```
(e / eat-01
   :ARG0 (d / dog)
   :ARG1 (b / bone))
```



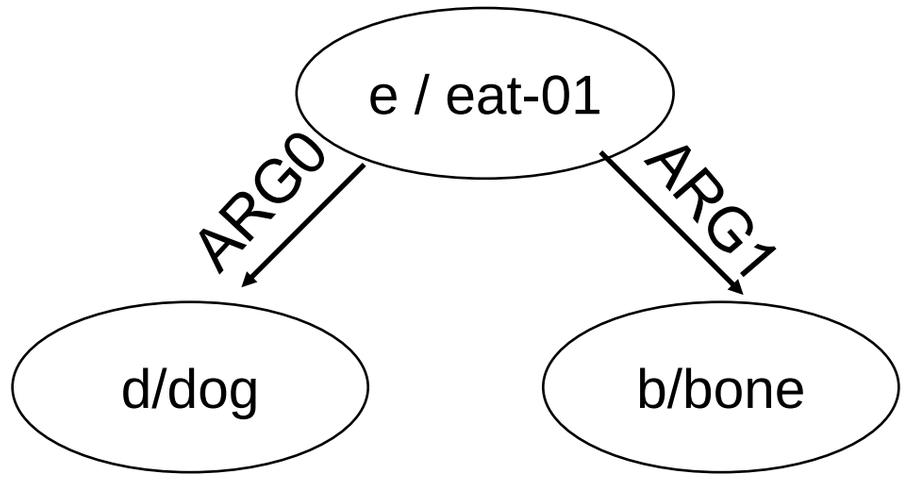
- Edges are labeled with **relations (including semantic roles)**
- Each node has a **variable.**
- Nodes are labeled with **concepts.**

- PropBank framesets used wherever possible.

AMR Example

The dog is eating a bone.

```
(e / eat-01
   :ARG0 (d / dog)
   :ARG1 (b / bone))
```



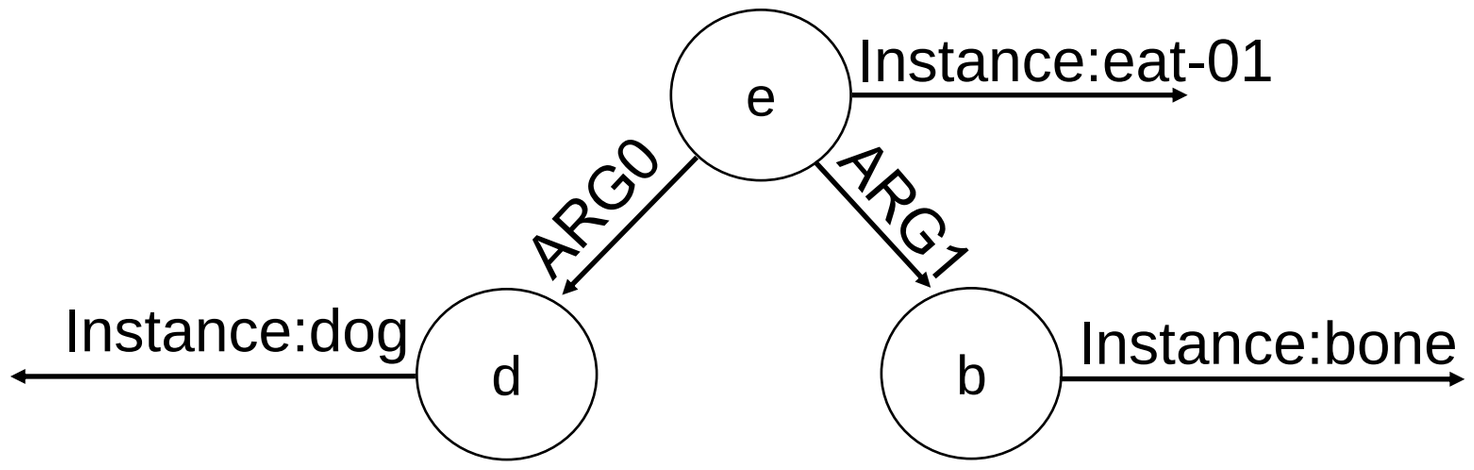
- Edges are labeled with **relations (including semantic roles)**
- Each node has a **variable.**
- Nodes are labeled with **concepts.**

- PropBank framesets used wherever possible.

AMR Example

The dog is eating a bone.

```
(e / eat-01
   :ARG0 (d / dog)
   :ARG1 (b / bone))
```

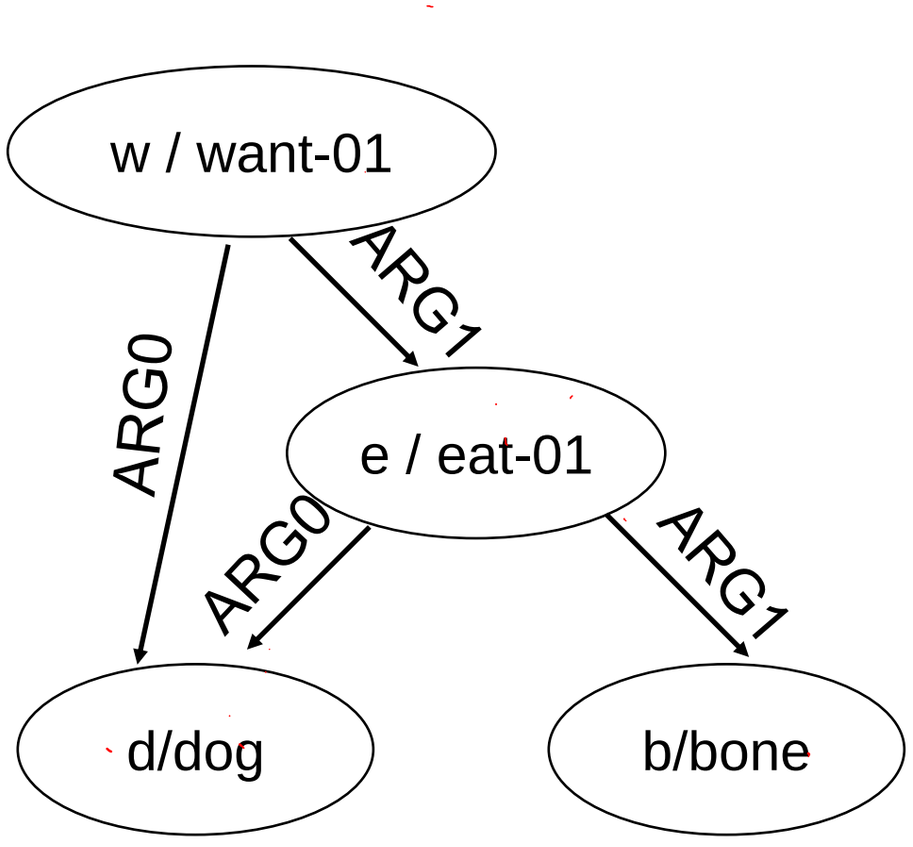


- Edges are labeled with **relations (including semantic roles)**
- Each node has a **variable.**
- Nodes are labeled with **concepts.**
  - Concepts can also be represented as edges.
- PropBank framesets used wherever possible.

Reentrancy

The dog wants to eat a bone.

```
(w / want-01
    :ARG0 (d / dog)
    :ARG1 (e / eat-01
        :ARG0 d
        :ARG1 (b / bone))
```

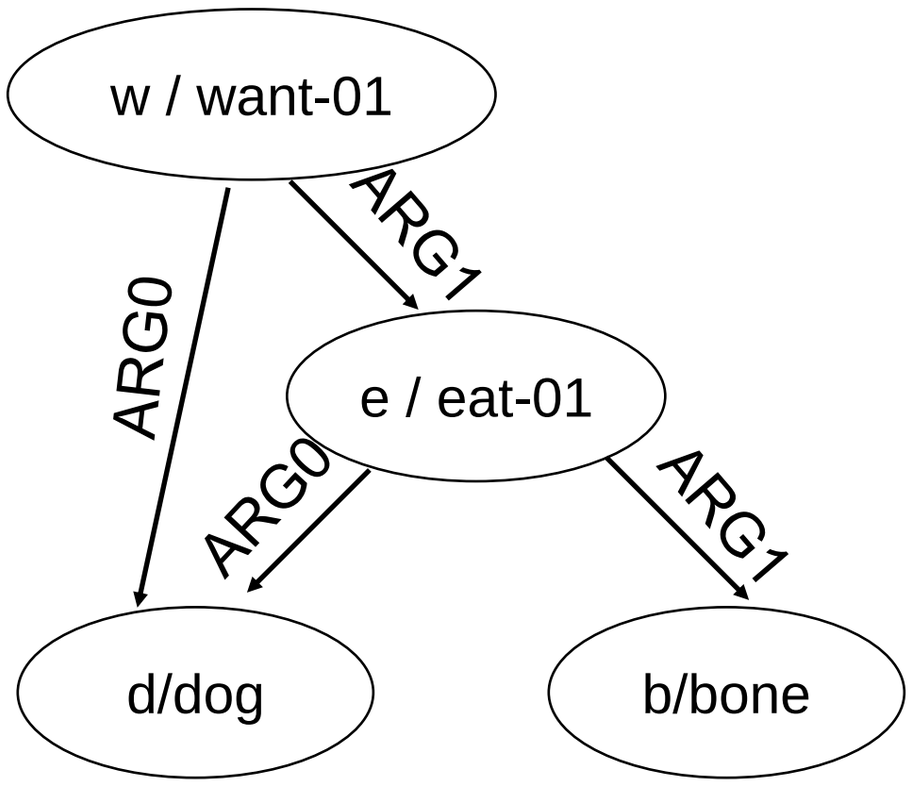


- Why the graph representation? Entities can play multiple roles.
- Two incoming edges in the graph, re-used variable in string notation.

AMR and Event Logic

The dog wants to eat a bone.

```
(w / want-01
    :ARG0 (d / dog)
    :ARG1 (e / eat-01
        :ARG0 d
        :ARG1 (b / bone))
```



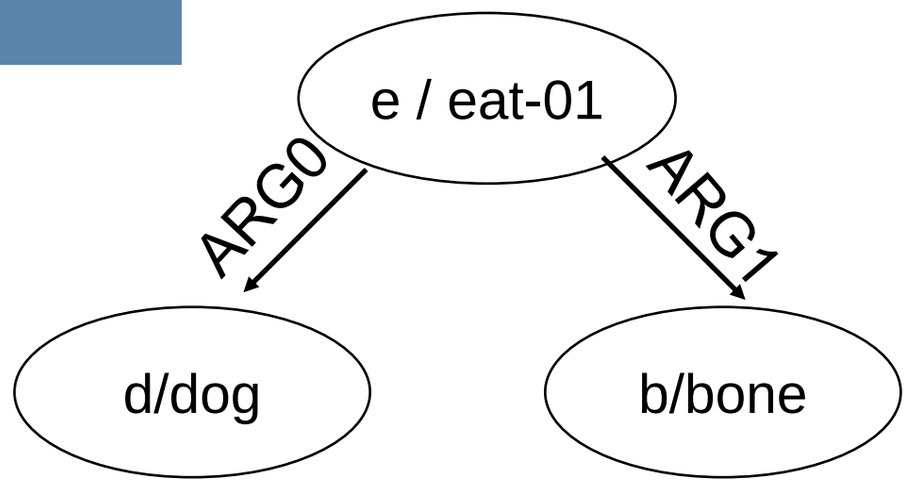
- AMR is related to event logic:
  - All concepts are existentially quantified.
  - Relations and concept labels are predicates.

$$\exists w\ \exists d\ \exists e\ \exists b\ Want(w) \wedge Dog(e) \wedge Eat(e) \wedge Bone(b) \wedge ARG0(w,d) \wedge ARG1(w,e) \wedge ARG0(e,d) \wedge ARG1(e,b)$$

Canonical Representaiton

The dog is eating a bone.
The bone was eaten by the dog.
The dog's eating of the bone.
...

```
(e / eat-01
   :ARG0 (d / dog)
   :ARG1 (b / bone))
```

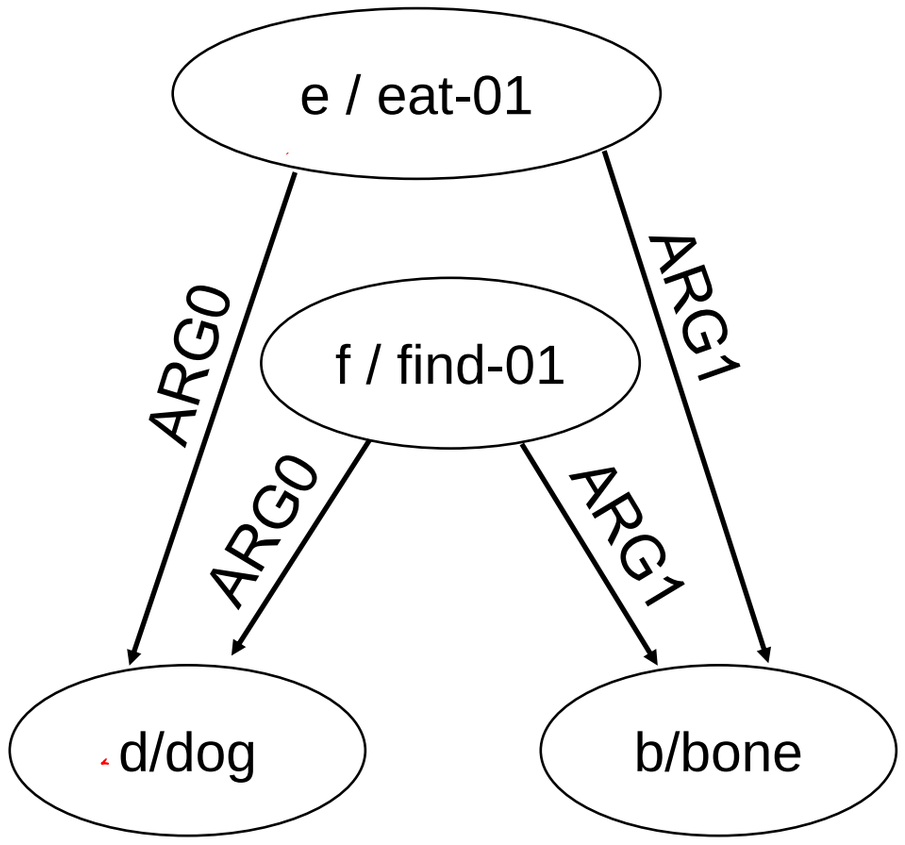


- Many different sentences can have the same AMR representation.
- Nouns can describe events too.

Inverse relations

The dog ate a bone that he found.

```
(e/ eat-01
   :ARG0 (d / dog)
   :ARG1 (b / bone))
(f/ find-01
   :ARG0 d
   :ARG1 b)
```

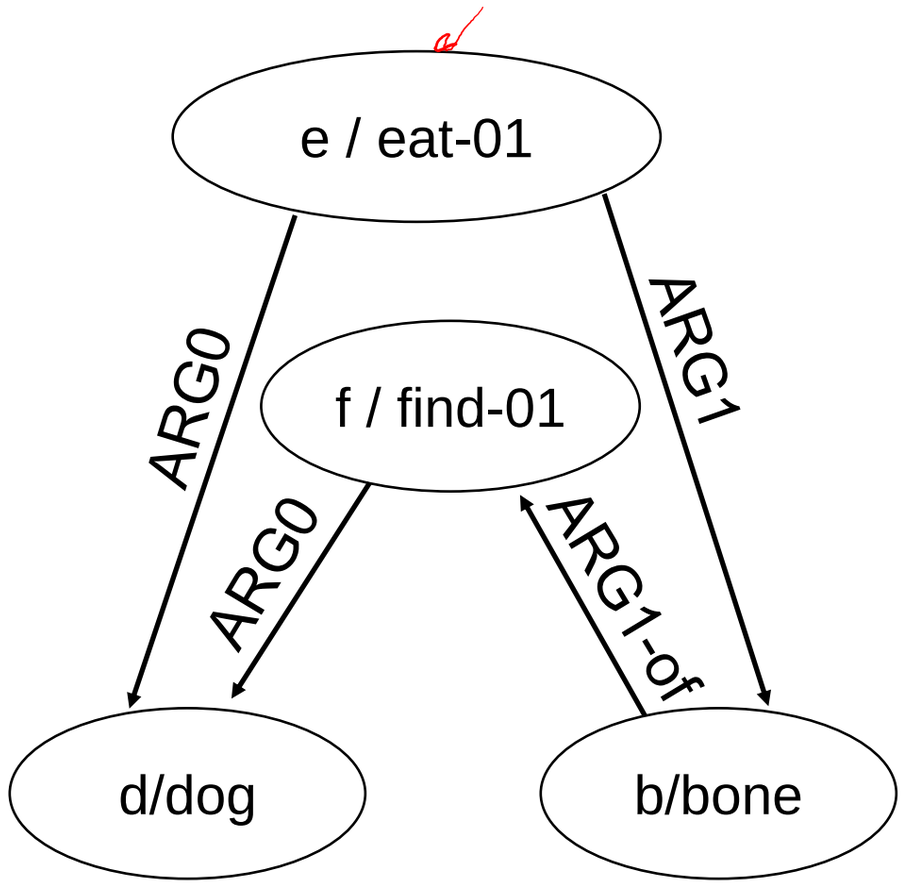


- AMR annotations are typically single-rooted (tree plus reentrancy)
- The single root is the "focus" of the sentence.

Inverse relations

The dog ate a bone that he found.

```
(e/ eat-01
    :ARG0 (d / dog)
    :ARG1 (b / bone
            :ARG1-of (f / find
                        :ARG0 d))
```

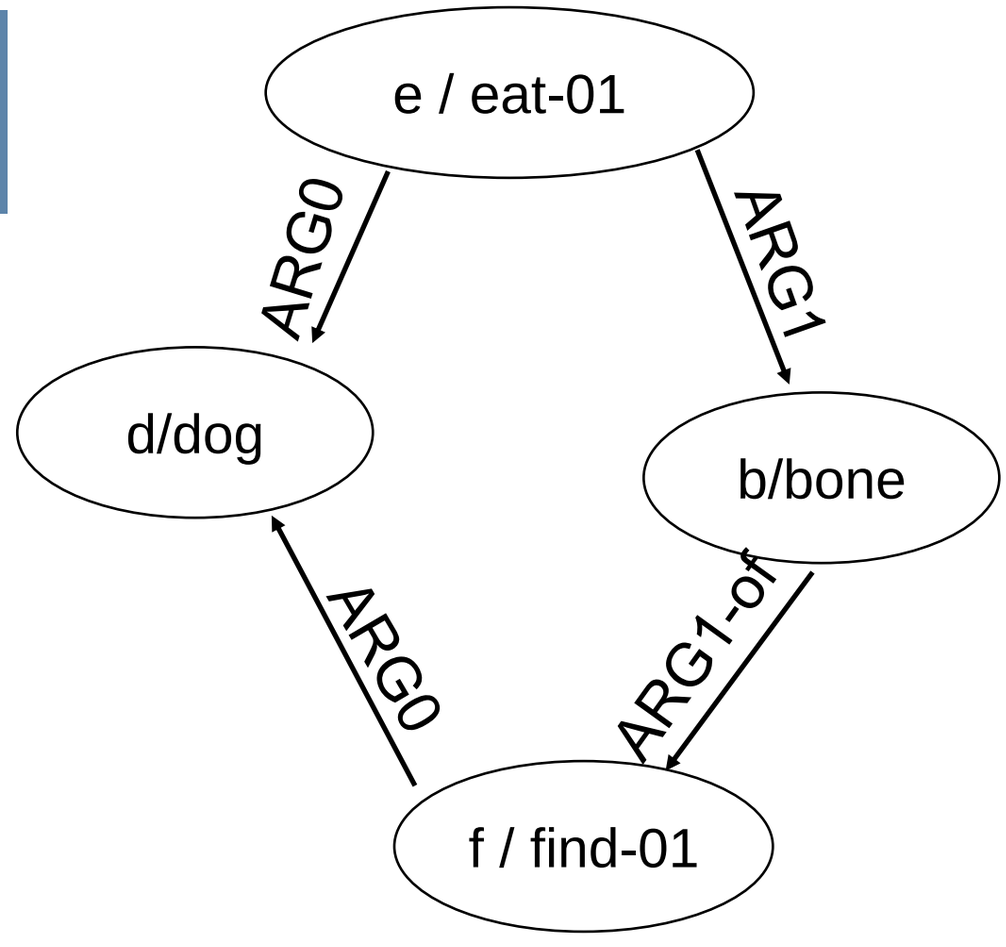


- AMR annotations are typically single-rooted (tree plus reentrancy)
- The single root is the "focus" of the sentence.

Inverse relations

The dog ate a bone that he found.

```
(e/ eat-01
    :ARG0 (d / dog)
    :ARG1 (b / bone
            :ARG1-of (f / find
                        :ARG0 d))
```

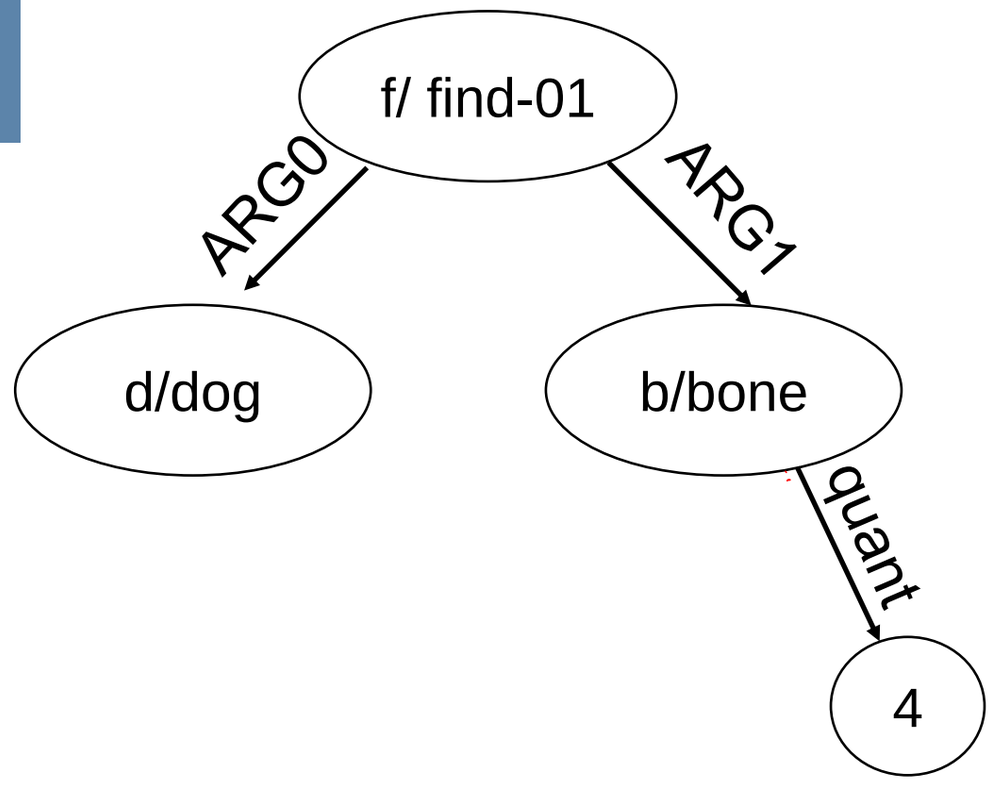


- AMR annotations are typically single-rooted (tree plus reentrancy)
- The single root is the "focus" of the sentence.

Constants

*The dog found **four** bones.*

```
(f/ find-01
    :ARG0 (d / dog)
    :ARG1 (b / bone
             :quant 4))
```

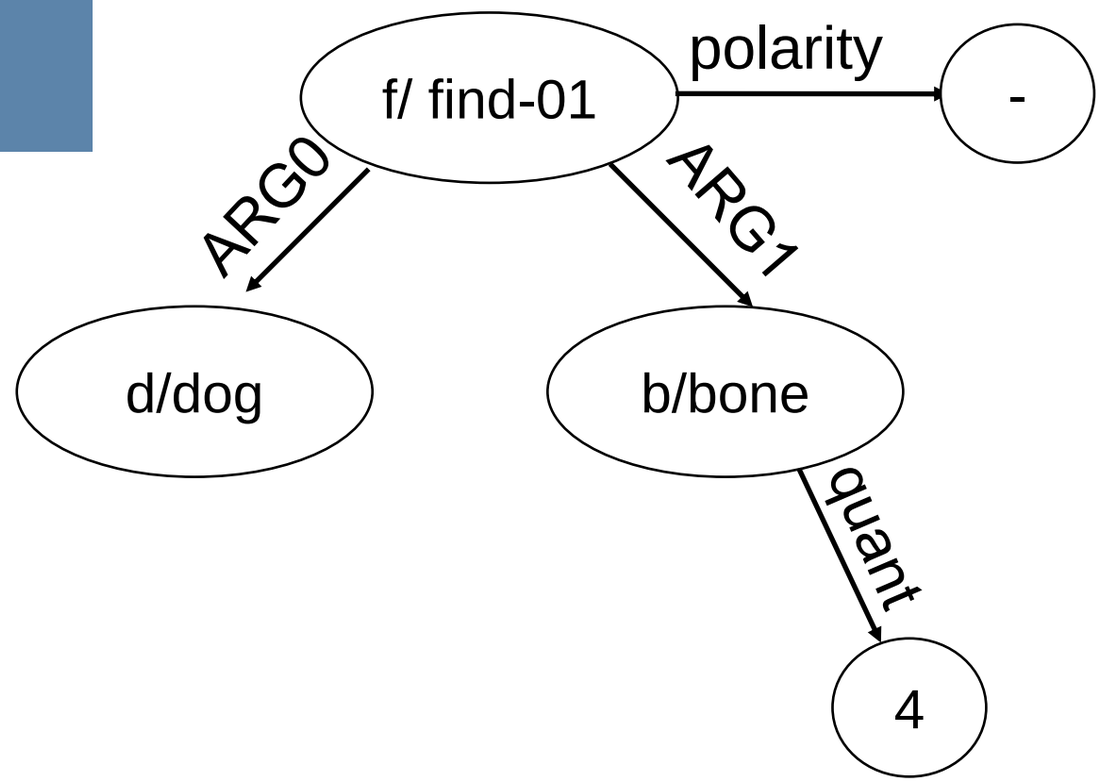


- Constants are used to represent quantities (node gets no variable).
- Also used for negation.

Constants

*The dog did not find **four** bones.*

```
(f/ find-01
   :ARG0 (d / dog)
   :ARG1 (b / bone
          :quant 4)
   :polarity -)
```



- Constants are used to represent quantities (node gets no variable).
- Also used for negation.

Non-Core Roles

- AMR annotations use some built-in relations (not in PropBank)
  :time, :location, :manner, :part, :frequency
- :mod and :domain for attributes
- :op1, op2, ...for lists of arguments (for example in conjunctions).

```
(t/ truck
    :mod (m / monster))
```

a monster truck.

```
(s/see-01
  (y / yummy
       :domain(f / food))
```

seeing that the food is yummy.

```
  (a / and
     :op1 (a / apple)
     :op2 (o / orange))
```

apples and oranges.

Names and Dates

```
(j / join-01
      :ARG0 (p / person :wiki -
            :name (p2 / name :op1 "Pierre" :op2 "Vinken")
            :age (t / temporal-quantity :quant 61
                  :unit (y / year)))
      :ARG1 (b / board
            :ARG1-of (h / have-org-role-91
                  :ARG0 p
                  :ARG2 (d2 / director
                        :mod (e / executive :polarity -))))
      :time (d / date-entity :month 11 :day 29))
```

AMR to English

```
(r / read-01
      :arg0 (j / judge)
      :arg1 (t / thing
                  :arg1-of (p /propose-01))
```

```
(p / picture-01
  :ARG0 (i / it)
  :ARG1 (b2 / boa
            :mod (c / constrictor)
            :ARG0-of (d / digest-01
                        :ARG1 (e / elephant))))
```

English to AMR

- *"The girl wants the boy to like her"*
- *"The girl wants the boy to believe that she likes him"*

AMR Data

- The Little Prince
  (publicly available, http://amr.isi.edu/download.html):
  - English and Chinese
  - Biomedical Data
- "AMRBank", 14k sentence, PTB and other corpora
  (including online discussion forums)

Another AMR Example

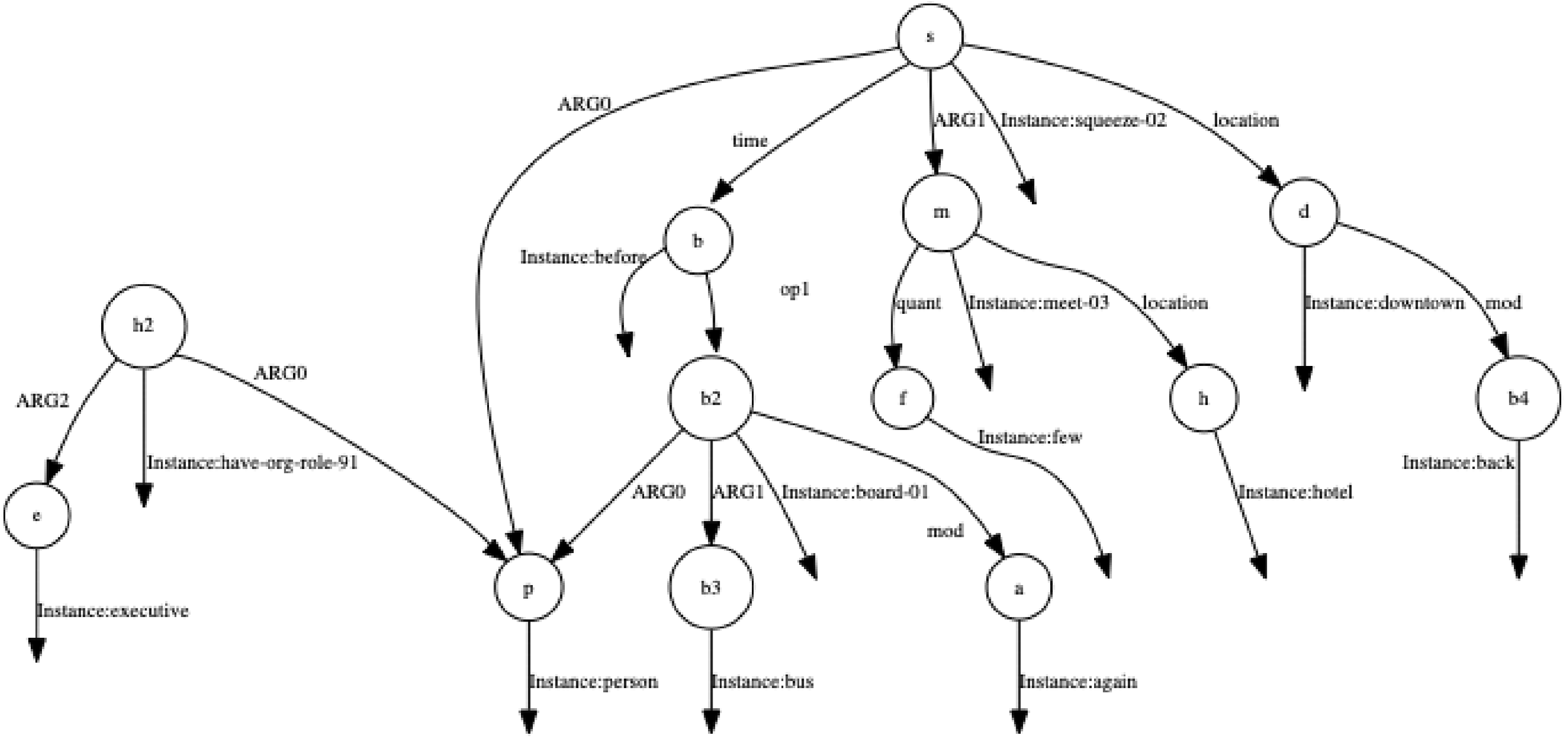
```
(s / squeeze-02
   :ARG0 (p / person
            :ARG0-of (h2 / have-org-role-91
                         :ARG2 (e / executive)))

   :ARG1 (m / meet-03
            :location (h / hotel)
            :quant (f / few))

   :location (d / downtown
                :mod (b4 / back))
   :time (b / before
            :op1 (b2 / board-01
                     :ARG0 p
                     :ARG1 (b3 / bus)
                     :mod (a / again))))
```

Back downtown, the execs squeezed in a few meetings at the hotel before boarding the buses again.

Another AMR Example



Back downtown, the execs squeezed in a few meetings at the hotel before boarding the buses again.

Applications of AMR

- Semantics-Based Machine Translation
  (Jones, Andreas, Bauer, Hermann & Knight, 2012)

- Summarization:
  - Abstractive Summarization
    (Liu, Flanigan, Thomson, Sadeh & Smith, 2015)
  - Text Compression (text-to-text generation)
    (Thadani, 2015)

- Predicting stock price movement from financial news
  (Xie, 2015)

Acknowledgments

- Some slides from Nathan Schneider & Jeff Flanigan's AMR tutorial at NAACL 2015.