Introduction to Abstract Meaning Representation AMR 2 – Writing AMRs

Bernard ESPINASSE
Aix-Marseille Université (AMU)
LIS UMR CNRS 7020, Marseille, France.

May, 2019

- Introduction to AMR
- AMR more in detail
- Writing AMRs
- Annotate AMRs
- Advanced AMR notation uses

References (1)

Books, articles and reports :

- Banarescu L., Bonial C., Cai S., Georgescu M., Griffitt K., Hermjakob U., Knight K., Koehn P., Palmer M., and Schneider N. (2013). Abstract Meaning Representation for Sembanking. In Proceedings of the 7th Linguistic Annotation Workshop and Interoperability with Discourse, 178–186, Sofia, Bulgaria: Association for Computational Linguistics. https://amr.isi.edu/a.pdf
- AMR 1.1 specification: http://www.isi.edu/ ulf/amr/help/amr-guidelines.pdf
- Palmer M. et al, The Proposition Bank: An Annotated Corpus of Semantic Roles, Comp. Linguistics, 31(1), 1-36, (2005).
- Banarescu L., Bonial C., Cai S., Georgescu M., Griffitt K., Hermjakob U., Knight K., Koehn P., Palmer M. & Schneider N. (2012). Abstract meaning representation (amr) 1.0 specification. In *Parsing on Freebase from Question-Answer Pairs."*
- Migueles-Abraira N., A Study Towards Spanish Abstract Meaning Representation, Master Thesis, 2017, University of the Basque Country.
- …

Courses/tutorials:

- N. Schneider, J. Flanigan, T. O'Gorman, "The Logic of AMR: Pratical, Unified Graph-Based Sentence Semantics for NLP", Tutorial at the 2015 Conference of the North American Chapter of the Association for Computational Linguistics.
- …

Outline (1)

1. Introduction to AMR

- **Computational semantic, syntactic and semantic analysis**
- **AMR formats and concepts**
- **What AMR represents and do not represents**

Computational semantic & syntactic analysis

- **Computational semantics** (Blackburn and Bos, 2013) :

  *discipline that combines insights from **formal semantics**, **computational linguistics**, and **automated reasoning** whose goal is to **construct semantic representations** for expressions of human language in an **automatic way**.*

- The **meaning of a sentence** depends so closely on its **syntactic structure**
- **Syntactic analysis** and **syntactic parsers** play an important role in representing such meaning :
  - Syntaxtical phrase structure helps to identify the **semantic relationship** that a *predicate* has with its given *arguments* in the description of a situation – also known as **semantic roles**

But syntactic analysis is not able to represent meaning

Limitation of syntactic analysis

- Given 4 sentences (from Matthews as cited in Chomsky, 1996):

  (1) The window broke

  (2) A hammer broke the window

  (3) The workman broke the window with a hammer

  (4) The window broke with a hammer

- Syntactically speaking, ***window*** is represented:
  - as the *verb's subject* in (1) and (4)
  - as the *verb's direct object* in the (2) and (3)
- All these sentences indicate that there is a broken thing: the window.
- However, a syntactic analysis is not able to depict this.

Who did what to whom, how, when, where, why, and with what consequences?

From syntactic analysis to semantic analysis

Syntactic analysis :

- **Completely assumed** by many statistical parsers trained on manually annotated syntactic database of sentences often in the form of a tree (Treebank)
- The **accuracy** of state-of-the-art syntactic parsers is **around 90%.**
- One of the most well-known English language treebanks is the Penn Treebank (PTB) (Marcus et al., 1993).

Semantic analysis :

- **Not currently assumed**
- Main reason : semantic annotation is « balkanized » (Banarescu et al., 2013), divided into separate annotations.
- **Lack of an unified model** to integrate various kinds of annotation data.
- Initiative towards a **graph-based parsing** for a more direct semantic analysis of whole natural language sentences : **AMR**

Introduction to AMR (1)

- **AMR** stands for **Abstract Meaning Representation** is a **contribution to Semantic Analysis**
- **AMR** concept firstly introduced in 1998 (Langkilde & Knight, 1998)
- **AMR** is a ***semantic representation language*** based on the assumption that we lack a **simple readable semantic bank** – or **sembank** – of natural language sentences "paired with their whole-sentence, logical meanings" (Banarescu et al., 2013)
- **AMR** is a **graph-based annotation language, encoded as** Rooted Directed Acyclic Graph permiting **rapid human annotation of corpora** with broad coverage
- **AMR** have to permit **ultimate advances in NLP tasks** : Statistical Natural Language Understanding, Statistical Machine Translation, …
- **AMR deals with** : discourse connectives, semantic roles, intrasentential coreference, named entities (wikification), questions, negation, & modality…

AMR formats (1)

- Very roughly AMR it is supposed to represent '**who** is doing **what** to **whom**, **where**, **when** and **how**' in a sentence S.
- AMR permits to abstract away meaning from syntactical representations, in the sense that sentences which are similar in meaning should be assigned the **same AMR**, even if they are not identically worded
- AMR have 3 equivalent formats:
  - **Logic format** : a formal representation
  - **AMR format**: a textual linearization based on Penman notation (Matthiessen et al., 1991), easy for human reading and writing
  - **Graph format**: for visualisation and used by programs

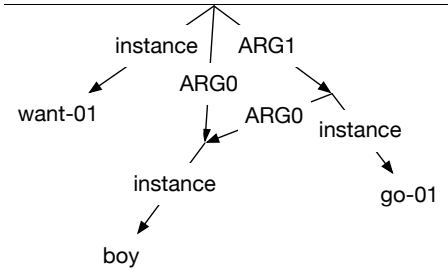
3 AMR formats (2)

Sentence : « *The boy wants to go* »

Logic formulation :

∃ w, b, g: instance(w, want-01) ∧ instance(g, go-01) ∧ instance(b, boy) ∧ arg0(w,b) ∧ arg1(w, g) ∧ arg0(g, b)

Graph formulation (DAG)



AMR formulation (based on PENMAN)

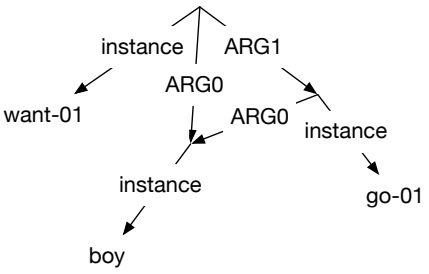
```
(w / want-01
   :arg0 (b / boy)
   :arg1 (g / go-01)
         arg0 b)
)
```


3 AMR Graph: alternative formulations

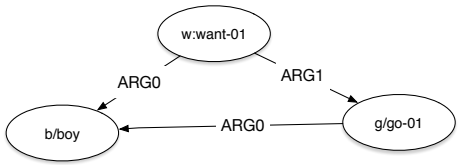
Sentence : « *The boy wants to go* »

∃ w, b, g: instance(w, want-01) ∧ instance(g, go-01) ∧ instance(b, boy) ∧ arg0(w,b) ∧ arg1(w, g) ∧ arg0(g, b)

Graph formulation 1



Graph formulation 1



AMR concepts (1)

Every AMR has a single **root** node at the top of the graph, which is considered to be the **focus**

Each **node** in the graph

- has a *variable* and represents a *semantic concept* (variable = instance of concept) a slash (/)
- variables are reused if something is referenced multiple times: *re-entrancy*
- *Semantic concepts* include PB framesets and English words

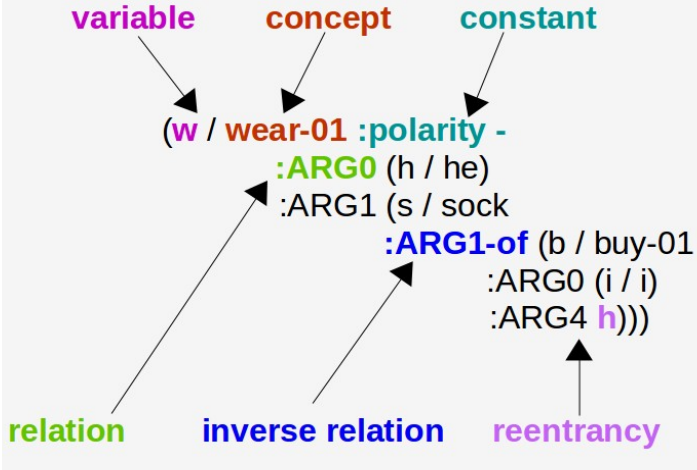
AMR concepts (2)

Graph **edges** denote *relations* between concepts

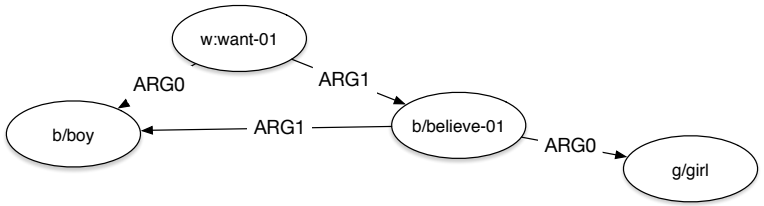
- *Semantic relations* include different types of **roles**, marked by a colon prefix (:)
- Some relations known as *constants* – get no variable, just a **value**
- Relation (role) can be *inverted* (useful for maintaining a single rooted structure)
- It is also possible to **convert** a *role* into a *concept* by reification (usefull to make a relation the focus of an AMR fragment)
- Yet not all *relations* have reifications

AMR concepts (3)

Sentence: "*He is not wearing the socks that I bought him*"



Sentence to AMR: many to one



The graph corresponds to all the following sentences:

- *The boy wants the girl to believe him.*
- *The boy wants to be believed by the girl.*
- *The boy has a desire to be believed by the girl.*
- *The boys desire is for the girl to believe him.*
- *The boy is desirous of the girl believing him.*

amongst others …

What AMR represent …

To capture many aspects of meaning in a single simple data structure AMR:

- abstracts away from *morpho-syntactic idiosyncrasies*
- focus on *logic* rather than in *syntactic representation*

AMR uses ***PropBank*** *(PB) framesets* (Palmer et al., 2005) :

- each frame presents *annotators* with a list of *senses*
- each *sense* has its *own definition* and its *own numbered arguments* (ARG)

AMR uses of approximately a ***100 semantic relations*** (*semantic roles*) organised in role categories

AMR *does not dictate imperative modelling rules*, but promotes *personal interpretation* about how strings are related to meanings.

What AMR do not represent …

In order to :

- obtain a *simple representation*
- to assign the *same AMR* to sentences that have the *same basic meaning*

AMR :

- does not represents *tense* and *number* of *verbs*
- do not represents *word category* and *order of words*
- drops in a sentence :
  - articles
  - most prepositions except for time and location prepositions

- AMR is not an interlingua, it is biased towards English.

Why drop articles?

- All mentions of a term go to **the same variable**, including **pronouns** and **later nominal mentions**

*I saw **a nice dog** and noticed **he** was eating a bone*

(d / dog
:mod nice)

Is "d" indefinite or definite?

- We do capture **demonstratives**:
- *Ex : This house*

(h / house
:mod (t / this))

Why drop tense ?

- English verbal tense **doesn't generalize well cross-linguistically**; not available for nominal predicates.
- Richer time representation might have required looking beyong a sentence
- Keep a simple representation.
- Example :

The man described the mission as a disaster.
The man's description of the mission: disaster.
As the man described it, the mission was a disaster.
The man described the mission as disastrous.

Same notation :

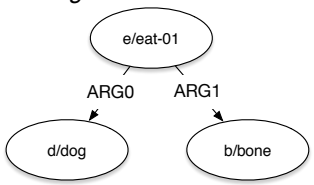
(d / describe-01
:arg0 (m / man)
:arg1 (m2 / mission)
:arg2 (d / disaster))

2. AMR more in detail

- **Penman notation and PropBank (PB)**
- **Concepts and Relations in AMR**
- **Inverse relation, negation, and questions in AMR**
- **Nouns, named entities in AMR**
- **Concepts versus constants**

Penman notation (1)

- Proposed by (Matthiessen et al., 1991)
- Sentence : « *The dog is eating a bone* »

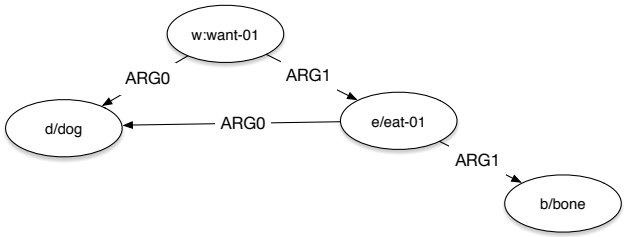


- Edges of the graph are relations : arg0 and arg1
- Each node in the graph has a variable labeled by concept :
  - d /dog means "d is variable instance of concept dog"
  - b /bone means "b is variable instance of concept bone"
  - e /eat-01 means "e is variable instance of concept (verb) eat-01"
- Ex:

```
(e / eat-01
    :arg0 (d / dog)
    :arg0 (b / bone))
```


Penman notations (2)

- Sentence : « *The dog wants to eat the bone* »



- **dog** has now 2 incoming roles
- In Penman we repeat the variable : we call this a **reentrancy** :

```
(want-01
    :arg0 (d / dog)
    :arg1 (e / eat-01
        :arg0 d
        :arg1 (b / bone)))
```


PropBank (PB)

Martha Palmer, Paul Kingsbury, Daniel Gildea, « **The Proposition Bank: A Corpus Annotated with Semantic Roles** », Computational Linguistics Journal, 31:1, 2005.

- **PropBank** is a **verb-oriented resource**, which generalizes descriptions across similar verbs (e.g. "describe" and "characterize") as well as nouns and other words (e.g. "description").
- **PropBank** permits to annotating all verbs in a corpus,
- **PropBank** does not annotate events or states of affairs described using nouns.
- **PropBank**-style annotations often remain close to the syntactic level,
- **PropBank** was developed with the idea of serving as training data for machine learning-based semantic role labeling systems
- **PropBank** requires that all arguments to a verb be syntactic constituents and different senses of a verb are only distinguished if the differences bear on the arguments.

AMR is largely based on **PropBank**

Concepts in AMR

AMR concepts are :

- **Words** :
  - e.g. boy, bone, dog, …
- **PropBank framesets** :
  - e.g. want-01, …
- **Special keywords** that include :
  - **entity types** :
    - e.g. date-entity
  - **quantities** :
    - e.g. distance-quantity, monetary quantity
  - **logical conjunctions** :
    - e.g. and
  - **…**

Semantic relations in AMR (1)

Types of relations: :

- Relations for **quantities** :

  :quant, :unit, :scale.

- Relations for **date-entities**:

  :day, :month, etc.

- Relations for **lists** :

  :op1 to :op10.

- It includes the **inverses** of all these relations :

  - :arg0-of.

Semantic relations in AMR (2)

AMR uses approximately **100 semantic relations** organised in role categories :

- core ":ARGX" roles (frame arguments)
- non-core roles (general semantic relations)
- roles for quantities
- roles used in date-entity
- roles of the form ":opX"
- roles of the form ":prep-X"
- multi-sentence roles
- conjunction role

Simple roles often correspond to a *reified concept* or *concepts*.

AMR Relations : PropBank (PB)

- Approximatively **100 relations** in **PropBank (PB)**
- **Frame arguments** using **PropBank framesets**, creating a frame for **verbs** with **arguments labelled**:

  :arg0 to :arg5 (or :a0 to :a5).

  :arg0 is typically the *agent*,

  :arg1 is typically the *patient*.

- **Other arguments** do not have standard definitions and may vary with the verb being annotated.

AMR Relations: general semantic relations

- **General semantic relations as :**
  - :accompanier, :age, :beneficiary,
  - :cause, :compared-to, :concession,
  - :condition, :consist-of, :degree,
  - :destination, :direction, :domain,
  - :duration, :employed-by, :example,
  - :extent, :frequency, :instrument,
  - :location, :manner, :medium, :mod,
  - :mode, :name, :part, :path, :polarity,
  - :poss, :purpose, :source, :subevent,
  - :subset, :time, :topic, :value, …

PropBank Frame-set: Example

Example : Frameset accept.01 « take willingly » :

The Proposition Bank — Palmer et al.

(12) Frameset **accept.01** "take willingly"
Arg0: Acceptor
Arg1: Thing accepted
Arg2: Accepted-from
Arg3: Attribute
Ex:[$_{Arg0}$ He] [$_{ArgM\text{-}MOD}$ would][$_{ArgM\text{-}NEG}$ n't] *accept* [$_{Arg1}$ anything of value] [$_{Arg2}$ from those he was writing about]. (wsj_0186)

Inverse AMR Relations: ARG-of

Use of ARG-of :

The boy from the college sang
```
(s / sing-01
   :arg0 (b / boy
      :source (c / college)
   )
)
```
The college boy who sang
```
(b / boy
   : arg0-of (s / sing-01)
   :source (c / college)
)
```
The number of pandas increased
```
(i / increase-01
   : arg1 (n / number
      :quant-of (p / panda)
   )
)
```


Negation (1)

The boy did not go
```
(g / go-01
   :arg0 (b / boy)
:polarity - )
```

The boy cannot go
```
(p / possible
   :domain (g / go-01
      :arg0 (b / boy)
   )
:polarity - )
```

Its possible for the boy not to go
```
(p / possible
   :domain (g / go-01
      :arg0 (b / boy)
      :polarity -
   ))
```


Negation (2)

The boy doesnt have to go.
The boy isnt obligated to go.
The boy need not go.

```
(p / obligate-01
   :arg2 (g / go-01
      :arg0 (b / boy)
   )
:polarity - )
```

The boy must not go

```
(p / obligate-01
   :arg2 (g / go-01
)
:arg0 (b / boy)
:polarity -)
```

Questions in AMR

In AMR we use « **amr-unknown** » to indicate what, who, when, … questions :

What *did the girl find?*

```
(f / find-01
   :arg0 (g / girl)
   :arg1 (a / amr-unknown))
```

Where *did the girl find the boy?*

```
(f / find-01
   :arg0 (g / girl)
   :arg1 (b / boy)
   :location (a / amr-unknown))
```


ArgM and modifier tags

In addition to the roles **Arg0** to **Arg5,** verbs can have modifier tags marked by **ArgM** together with a **modifier tag** from this list:

- LOC: location,
- EXT: extent,
- DIS: discourse connectives,
- ADV: general purpose,
- NEG: negation marker,
- MOD: modal verb,
- CAU: cause,
- TMP: time,
- PNC: purpose,
- MNR: manner,
- DIR: direction

Polysemous verbs (PropBank)

- Polysemous verbs have multiple frames (PropBank) :

Frameset cut.01 'reduce'
Arg0: cutter
Arg1: thing cut
Arg2: medium, source
Arg3: instrument
Ex: [Arg0 Longer production runs] [ArgM-MOD would] cut [Arg1 inefficiencies from adjusting machinery between production cycles]. (wsj-0317)

Frameset cut.04 cut off = slice
Arg0: cutter
Arg1: thing cut (off)
Arg2: medium, source
Arg3: instrument
Ex: [Arg0 The seed companies] cut off [Arg1 the tassels of each plant]. (wsj-0209)

Nouns

- Some nouns are represented via AMRs.
- Example :

```
(d / destroy-01
   :arg0 (b / boy)
   :arg1 (r / room))
```

For the sentences :

the destruction of the room by the boy...

the boys destruction of the room ...

The boy destroyed the room.

Named Entities

- AMR allows any concept to be named via a named relation.
- AMR includes standard forms for approximatively 80 types like : person, country, ...

- Example :

Hari Kumar

(p / person
:name (n / name
:op1 "Hari"
:op2 "Kumar"))

Constants

- Some relations, called **constants**, get no **variable**.
- The editor does this **automatically** for certain contexts.

- ***negation*** : *"The dog **did not** eat the bone"*
  (e /eat-01 :polarity -
  :arg0 (d / dog)
  :arg1 (b / bone))

- ***numbers*** : *"The dog ate **four** bones"*
  (e /eat-01
  :arg0 (d / dog)
  :arg1 (b / bone **:quant 4**))

- ***names*** : *"**Fido** the dog"*
  (d / dog
  :name (n / name :op1 **"Fido"**))

Concepts vs. Constants

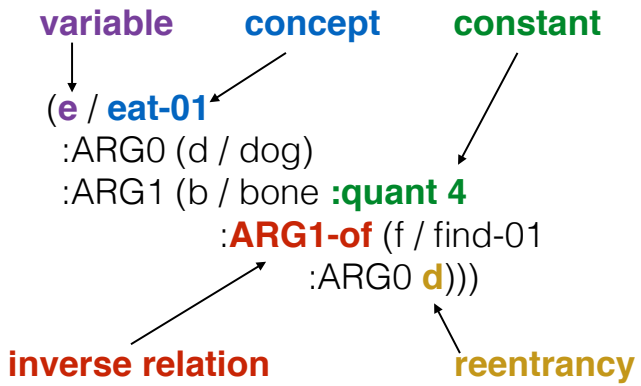
A **concept** is a type :

- For every concept node there will be ≥1 instance variable/node.
- An instance can be mentioned multiple times
- Multiple instances of the same concept can be mentioned.

Constants are singleton nodes:

- no variable, just a value.
- Specific non-core roles allow constant values.

Summary of AMR notation



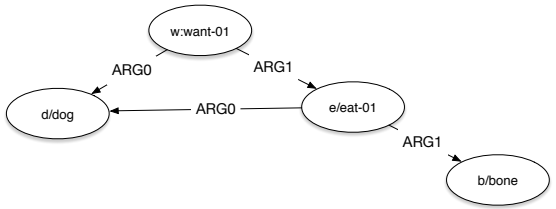
Now, **how we annotate AMRs** …

3. Writing AMRs

- **Reentrancy**
- **Focus & Inverse relation**
- **Reviewing the format**
- **AMR Parsing and Alignment**

Reentrancy

- Sentence : « *The dog wants to eat the bone* »



- It **does not matter** where the ***concept label*** goes :

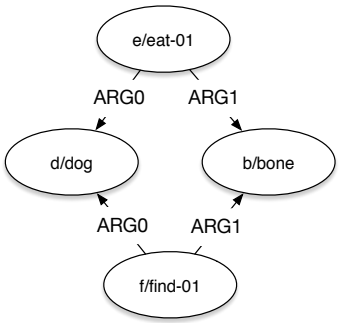
```
(want-01
   :arg0 (d / dog)
   :arg1 (e /eat-01
      :arg0 d
      :arg1 (b / bone)))
```

⇔

```
(want-01
   :arg0 d
   :arg1 (e /eat-01
      :arg0 (d / dog)
      :arg1 (b / bone)))
```


Focus (1)

- Sentence : « *The dog ate* ***the bone that he found*** »



- How do we know **what goes on top**?
- How do we get these **into the AMR format** ?

Focus (2)

- We call “what goes on top” the **focus**
- Conceptually, **the main assertion**
- Linguistically, **often the head**
- For a sentence, **usually the main verb**
- Examples :
  - The **man** at the hotel : focus = « man »
  - The **hotel** the man is at : focus = « hotel »
  - The dog **ran** : focus = « ran »
  - The **dog** that ran : focus = « dog »

Inverse relation & Focus

Examples of **focus** :

The **man** at the hotel	(m / man **:location** (h / hotel))
The dog **ran**	(r / ran-01 **:arg0** (d / dog))
The **hotel** the man is at	(h / hotel **:???** (m / man))
The **dog** that ran	(d / dog **:????** (r / ran-01))

Inverse relations :

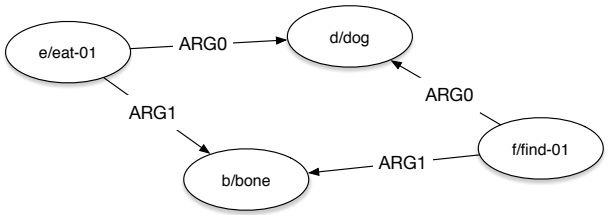
The **hotel** the man is at	(h / hotel **:location-of** (m / man))
The **dog** that ran	(d / dog **:arg0-of** (r / ran-01))

Notation convention :

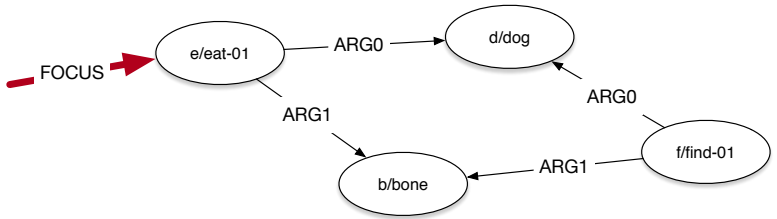
X arg0-of Y = Y arg0 X

Reviewing the format (1)

- Imagine a graph for "***The dog ate the bone that he found***" :



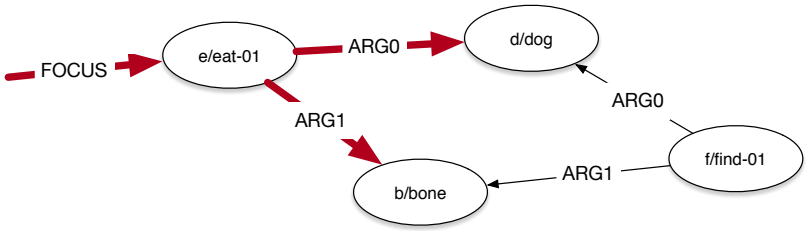
- **1) Find the focus :**



(e / eat-01 ...)

Reviewing the format (2)

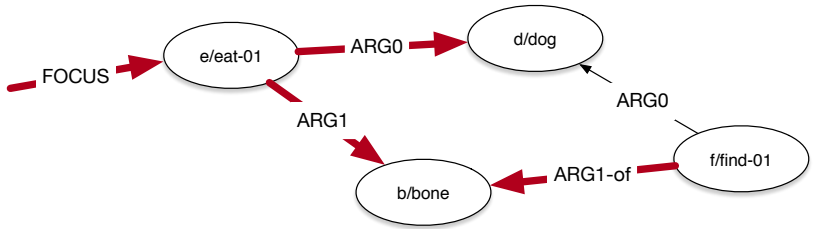
- **2) Add entities :**



(e/eat-01
:arg0 (d / dog)
:arg1 (b / bone))

Reviewing the format (3)

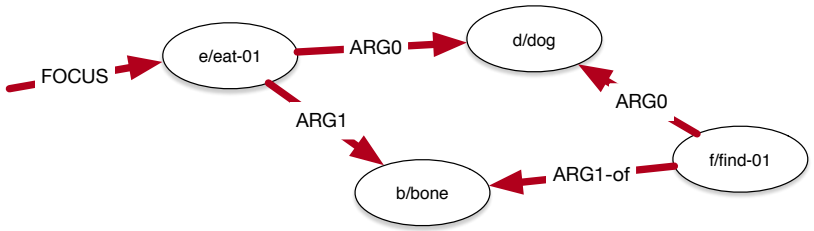
- **3) Invert a relation if needed:**



(e/eat-01
:arg0 (d / dog)
:arg1 (b / bone)
:arg1-of (f / find-01)))

Reviewing the format (4)

- **4) Add reentrancies:**




```
(e/eat-01
  :arg0 (d / dog)
  :arg1 (b / bone)
  :arg1-of (f / find-01
             :arg0 d)))
```


AMR parsing: from text to AMR: strategies

1. Graph-based AMR Parser:

- Separate parsing task into concept identification and relation identification
- Aim to find a connected graph with a maximum sum of edge (relation) scores

2. Transition-Based AMR Parser:

- Generate AMR graphs through conversion from dependency parse trees
- Design different parsing actions
- State-of-the-art system: **CAMR** (Wang et al. 2015a, 2015b) - ***F1 : 0.62***

But, no gold standard word-concept mappings …

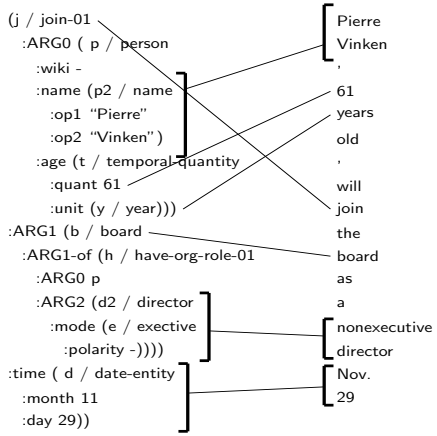
AMR alignment: strategies (1)

1. Align AMR concepts and relations to spans of words

"Pierre Vinken, 61 years old, will join the board as a nonexecutive director, nov. 29"

- **Heuristic Aligner :**
  - JAMR (Flanigan et al., 2014)
- **Unsupervised Aligner :**
  - ISI Aligner (Pourdamghani et al., 2014)
  - Stanford Aligner (Werling et al., 2015)

Source: Wei-Te Chen & Martha Palmer, 2017



AMR aligment: strategies (2)

2. Aligns AMR concepts and relations to word nodes in a dependency parse tree

"Pierre Vinken, 61 years old, will join the board as a nonexecutive director, nov. 29"

- **AMR concept → dependency parse node** (one-to-one) **alignment**
- Aim to **find better alignments** to benefit AMR parsing

Source: Wei-Te Chen & Martha Palmer, 2017

se tree