

Introduction to Abstract Meaning Representation

AMR 3 – Advanced AMR

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

May, 2019

- Annotate AMRs
- Advanced AMR notation uses
- Conclusion

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. Annotate AMRs**
 - PropBank Lexicon and AMR dictionary
 - Stemming Concepts
 - Core roles and non-core roles
 - Named entities and Wikification
 - Mesurables entities
- **2. Advanced AMR notation uses**
 - Typical uses of inverse roles
 - Compositionality criterion and relational nouns
 - Coordination and clausal connectives
 - Modal Concepts and Speech Acts
 - Sentence Types, questions, quantifications and sets
 - Reification
- **3. Conclusion**
 - Working with AMR Data sets
 - AMR Strength and Limitations
 - Further AMR documentation

1. Annotate AMRs

- **PropBank Lexicon and AMR Dictionary**
- **Stemming Concepts**
- **Core roles and non-core roles**
- **Named entities and Wikification**
- **Mesurables entities**

PropBank Lexicon

- Predicates use the **PropBank inventory**
- Each frame presents annotators with a list of senses
- Each sense has its own definitions for its numbered (core) arguments
- We generalize across **parts of speech** and **etymologically related words**:

My fear of snakes *fear-01*
I am fearful of snakes *fear-01*
I fear snakes *fear-01*
I'm afraid of snakes *fear-01*

- But we **don't** generalize over synonyms:
- My fear of snakes* *fear-01*
I'm terrified of snakes *terrify-01*
Snakes creep me out *creep_out-03*

run-01 - "operate, proceed, operate or proceed"

- ARG0: operator
- ARG1: machine, operation, procedure
- ARG2: employer
- ARG3: coworker
- ARG4: instrumental

Aliases: *run* (v), *run* (n), *running* (n)
[more](#)

run-02 - "walk quickly, a course or contest, run/jog, run for office"

- ARG0: runner theme
- ARG1: course, race, distance location
- ARG2: opponent

Aliases: *run* (v), *run* (n), *running* (n)
[more](#)

S

run-03 - "cost"

- ARG1: commodity
- ARG2: price
- ARG3: buyer

Aliases: *run* (v), *running* (n)

AMR dictionary

how

- **:ARGx**
 - **How much** does this cost? (*cost-01* :ARG2 amr-unknown) [Example](#)
 - **How did you solve the problem?** (*solve-01* :ARG2 amr-unknown) [Example](#)
- **:degree**
 - **How big** was the dog that bit you? (*big* :degree amr-unknown :domain :dog) [Example](#)
 - **How beautiful** you are. [= You are so beautiful.] (*beautiful* :degree so :domain you) [Example](#)
- **:manner**
 - **How did you get him to help you?** (*get-04* :manner amr-unknown) [Example](#)
 - The only thing that surprises me is **how rapidly** this is happening. (*surprise-01* :ARG0 (rapid :manner-of this)) [Example](#)
- **:quant**
 - **How tall** are you? (*tall* :quant amr-unknown :domain you) [Example](#)
 - **How old** is your father? (*father* :age amr-unknown :poss you) [Example](#)

however

- **contrast-01** (most cases)
 - John, however, stayed at home. (*contrast-01* :ARG2 stay-01) [Example](#)
- **:concession**
 - It started to rain; however, the game continued. (*start-01* :ARG1 rain-01 :concession-of continue-01) [Example](#)

imperative *⚡*

- **:mode imperative**
 - Go home! (*go-02* :mode imperative :ARG0 you :destination home) [Example](#)
 - Let's go home. (*go-02* :mode imperative :ARG0 we :destination home) [Example](#)
 - Come on folks!! (*come-25* :mode imperative :ARG1 folk) [Example](#)
- **:mode imperative :polite +**
 - Please close the door. (*close-01* :mode imperative :polite +) [Example](#)
 - Could you close the door? (*close-01* :mode imperative :polite +) [Example](#)

Stemming Concepts

- Non-predicates don't have PropBank frames : They are simply **stemmed**.
- All concepts drop **plurality, articles, and tense**:

A cat	Eating
The cat	Eats
Cats	Ate
The cats	Will eat
(c / cat)	(e / eat-01)

Pronouns that do not have a coreferent nominal mention are **made nominative** and **used as normal concepts**.

A man save himself	He saved himself	He save me
(s / save-01 :arg0 (m / man) :arg1 m)	(s / save-01 :arg0 (h / he) :arg1 h)	(s / save-01 :arg0 (h / he) :arg1 (i / i))

Go to the AMR Editor: <http://tiny.cc/amreditor>

Core Roles (1)

- If a semantic role is not in the **core roles** for a roleset, AMR provides an inventory of **non-core roles**

run-01 - "operate, proceed, operate or proceed"

- ARG0: operator
- ARG1: machine, operation, procedure
- ARG2: employer
- ARG3: coworker
- ARG4: instrumental

- **Non-core roles** express things like **:time, :manner, :part, :location, :frequency**

- **General semantic roles (incl. shortcuts):** [:accompanier](#) [ex](#) [:age](#) [ex](#) [:compared-to](#) [ex](#) [:concession](#) [ex](#) [:condition](#) [ex](#) [:consist-of](#) [ex](#) [:cos](#) [:direction](#) [ex](#) [:domain](#) [ex](#) [:duration](#) [ex](#) [:employed-by](#) [ex](#) [:example](#) [:instrument](#) [ex](#) [:li](#) [ex](#) [:location](#) [ex](#) [:manner](#) [ex](#) [:meaning](#) [ex](#) [:med](#) [:name](#) [ex](#) [:ord](#) [ex](#) [:part](#) [ex](#) [:path](#) [ex](#) [:polarity](#) [ex](#) [:polite](#) [ex](#) [:poss](#) [:source](#) [ex](#) [:subevent](#) [ex](#) [:subset](#) [ex](#) [:superset](#) [ex](#) [:time](#) [ex](#) [:topic](#) [ex](#)
- **In quantities:** [:quant](#) [ex](#) [:unit](#) [ex](#) [:scale](#) [ex](#) [:examples](#) [:quantity](#) [ty](#)
- **In date-entity:** [:day](#) [:month](#) [:year](#) [:weekday](#) [:time](#) [:timezone](#) [:year2](#) [:decade](#) [:century](#) [:calendar](#) [ex](#) [:era](#) [ex](#) [:mod](#) [:date-entity](#) [:era](#)
- **Ops:** [:op1](#) [:op2](#) [:op3](#) [:op4](#) [:op5](#) [:op6](#) [:op7](#) [:op8](#) [:op9](#) [:opl](#) [:op10](#)
- **In multi-sentence:** [:snt1](#) [:snt2](#) [:snt3](#) [:snt4](#) [:snt5](#) [:snt6](#) [:snt7](#) [:snt8](#)

- Inventory an handout, or in editor (the **[roles]** button)

Core Roles (2)

- Some relations need to have an ordered list of arguments, but **don't have specific meanings** for each entry.
- We use **:op1**, **:op2**, **:op3**, ... for these
- :op#** for coordination :
Apples and bananas
(a / and
:op1 (a2 / apple)
:op2 (b / banana))
- :op#** for names :
Barack Obama
(p / person
:name (n / name
:op1 "Barack" :op2 "Obama"))

Non-core Roles Inventory (1)

We use:

- :mod** for attribution
- :domain** is the inverse of **mod** (:domain = :mod-of)

Example:

<i>The yummy food</i> <i>There is yummy food</i> (f / food :mod (y / yummy))	<i>The yumminess of the food</i> <i>The food is yummy</i> (y / yummy :domain (f / food))
<i>seeing the yummy food seeing</i> <i>the food that is yummy</i> (s / see-01 :arg1 (f / food :mod (y / yummy)))	<i>seeing that the food is yummy</i> (s / see-01 :arg1 (y / yummy :domain (f / food)))

Non-core Roles Inventory (2)

This is also used for attributive/predicate demonstratives and nominals :

<i>This house</i> (h / house :mod (t / this))	<i>A monster truck</i> (t / truck :mod (m / monster))
<i>the truck is a monster</i> (m / monster :domain (t / truck))	

Named Entities (1)

Entities with names get special treatment!

Barack Obama

- (p / person
:name (n / name
:op1 "Barack"
:op2 "Obama"))

- We assign a **named entity type** from our ontology.
- 70+ categories like **person**, **criminal-organization**, **newspaper**, **city**, **food-dish**, **conference**!
- See your handout, or the **[NE types]** button in the editor
- Each gets a **:name** relation to a **name node**
- That node gets :op# relations to the strings of their name **as used in the sentence**.

Named Entities (2)

If there is a more specific descriptor present in the sentence, we use that **instead of the NE inventory**.

- « *a Kleenex* »

(p / product
:name (n / name
:op1 "Kleenex"))

- « *a Kleenex tissue* »

(t / tissue
:name (n / name :op1 "Kleenex"))

Wikification

In a second pass of annotation, we add **:wiki** relations.

- « *Barack Obama* »

(p / person
:name (n / name
:op1 "Barack"
:op2 "Obama")
:wiki Barack_Obama)

http://en.wikipedia.org/wiki/Barack_Obama

Measurable Entities

We also have special entity types we use for **normalizable entities**

- “Tuesday the 19th”

(d / date-entity
:weekday (t / tuesday)
:day 19)

- “five bucks”

(m / monetary-quantity
:unit dollar
:quant 5)

- “\$3 / gallon”

(r / rate-entity-91
:arg1 (m / monetary-quantity
:unit dollar
:quant 3)
:arg2 (v / volume-quantity
:unit gallon
:quant 1))

2. Advanced AMR notation uses

- Typical uses of inverse roles
- Compositionality criterion and relational nouns
- Coordination & clausal connectives
- Modal Concepts & Speech Acts
- Sentence Types, questions, quantifications and sets
- Reification

Various English Language Modelling Problems ...

- Copulas, light verbs
- Non-declarative sentences
- Questions, Comparisons & Quantification
- Prepositions
- Derivational morphology
- Relational nouns
- Coordination & clausal connectives
- Modality
- Subsets
- Reification
- ...

Light semantics

We try to eliminate purely grammatical words. E.g.:

- **copulas:** *I am happy*
(h / happy :domain (i / i))
- **light verbs:** *I'm taking a bath*
(b / bathe-01 :arg0 (i / i))
- **Heavy prepositions :** Most prepositions mark a (core or non-core) role. Some add crucial additional information meriting a concept:
- **at the school:** :location (s / school) **next to the school:**
:location (n / next-to :op1 (s / school))
- **between the school and the house:**
:location (b / between :op1 (s / school)
:op2 (h / house))
- **at the time of the war:** :time (w / war)
- **after the war:** :time (a / after :op1 (w / war))

Typical uses of inverse roles

- **Relative clauses:**
 - *someone who paints furnitures*
- **Derivational morphology :**
 - **Participles:** *furniture -painting person*
 - **Nominalizations:** *furniture painter*

(p / person
:arg0-of (s / paint-01
:arg1 (t / furniture)))

Concept « person » may be implicit ...

Compositionality criterion (1)

- We only “decompose” **derivational morphology** if a relative clause paraphrase is possible:
 - **teacher** = person who teaches
(p / person :arg0-of (t / teach-01))
 - **professor** ≠ person who professes
(p / professor)
- Often core roles are available for modifiers:
 - **math teacher** / **teacher of math** = person who teaches math
(p / person :arg0-of (t / teach-01
:arg1 (m / math)))
 - **math professor** ≠ person who professes math
(p / professor :mod (m / math))

Compositionality criterion (2)

- Sometimes it is difficult to draw a line, but we do our best:
 - **opinion** = thing that is opined
(t / thing :arg1-of (o / opine-01))
 - **profession** ≠ thing that is professed
(p / profession)

Hallucinating relations

- Sometimes we have to “hallucinate” a **relationship that the grammar underspecifies**.
 - e.g., **possessives** and **noun-noun compounds** can express many different kinds of relations

Relational nouns

- Special predicates for individual–group and individual–individual relations:
 - He is a pilot for TWA / He is a TWA pilot
(h / **have-org-role-91**
:arg0 (h2 / he)
:arg1 (c / company
:name (n / name :op1 "TWA"))
:arg2 (p / pilot))
 - I am your father
(h / **have-rel-role-91**
:arg0 (i / i)
:arg1 (y / you)
:arg2 (f / father))

Coordination & Clausal Connectives

Connectives	AMR notation
and	and
or	or
but	contrast-01
because ; due to ; on account of	:cause
(in order) to ; so (that)	:purpose
if	:condition
unless	:condition (... :polarity -)
although ; despite	:concession

Coordination

The most common patterns:

X, Y and Z	X, Y, or Z	X but Y
(a / and :op1 X :op2 Y :op3 Z)	(o / or :op1 X :op2 Y :op3 Z)	(c / contrast-01 :arg1 X :arg2 Y)

Coordination example (1)

Source: Nathan Schneider & Tim O’Gorman, 2018

« Rachael Ray finds inspiration in cooking her family and her dog. »

Step 1 :

```
(i / inspire-01
 :arg0 (c / cook-01
 ...
 :arg1 (p / person :name (n / name :op1 "Rachael" :op2 "Ray"))))
```

Step 2 :

```
(i / inspire-01
 :arg0 (c / cook-01
 :arg0 p
 :arg1 (a / and
 :op1 (f / family
 ...
 :op2 (d / dog :poss p )))
 :arg1 (p / person :name (n / name :op1 "Rachael" :op2 "Ray"))))
```

Coordination example (2)

Source: Nathan Schneider & Tim O’Gorman, 2018

« Rachael Ray finds inspiration in cooking her family and her dog. »

Step 3 :

```
(i / inspire-01
 :arg0 (c / cook-01
 :arg0 p
 :arg1 (a / and
 :op1 (f / family
 ...
 :op2 (d / dog :poss p )))
 :arg1 (p / person :name (n / name :op1 "Rachael" :op2 "Ray"))))
```

Coordination example (3)

Source: Nathan Schneider & Tim O’Gorman, 2018

« Rachael Ray finds inspiration in cooking her family and her dog. »

Step 4 :

```
(i / inspire-01
 :arg0 (c / cook-01
 :arg0 p
 :arg1 (a / and
 :op1 (f / family
 :arg1-of (h / have-org-role-91
 :arg0 p
 :arg2 (m / member)))
 :op2 (d / dog :poss p)))
 :arg1 (p / person :name (n / name :op1 "Rachael" :op2 "Ray"))))
```

X’s family = family of which X is a member

Coordination example (4)

Source: Nathan Schneider & Tim O’Gorman, 2018

« Rachael Ray finds inspiration in cooking her family and her dog. »

Step 5 :

```
(i / inspire-01
 :arg0 (c / cook-01
 :arg0 p
 :arg1 (a / and
 :op1 (c / cook-01
 :arg0 p)
 :op2 (f / family
 :arg1-of (h / have-org-role-91
 :arg0 p
 :arg2 (m / member)))
 :op3 (d / dog :poss p)))
 :arg1 (p / person :name (n / name :op1 "Rachael" :op2 "Ray"))))
```

Coordination: shared core args (1)

Source: Nathan Schneider & Tim O’Gorman, 2018

« We invited *and* then disinvited the students »

```
(a / and
:op1 (i / invite-01
      :arg0 (w / we)
      :arg1 (s / student))
:op2 (d / disinvite-01
      :arg0 w
      :arg1 s
      : time (t /then)))
```

Coordination: shared core args (2)

Source: Nathan Schneider & Tim O’Gorman, 2018

« Yesterday we invited *and* then disinvited the students »

```
(a / and
:op1 (i / invite-01
      :arg0 (w / we)
      :arg1 (w / student))
:op2 (d / disinvite-01
      :arg0 w
      :arg1 s
      : time (t /then))
      :time (y / yesterday))
```

« We invited the students *and* then the professors »

```
(a / and
:op1 (i / invite-01
      :arg0 (w / we)
      :arg1 (s / student))
:op2 (i2 / invite-01
      :arg0 w
      :arg1 p / professor)
      : time (t /then)))
```

Modal Concepts

Example 1:

You *can* leave.
You *may* leave
It’s all right for you to leave.

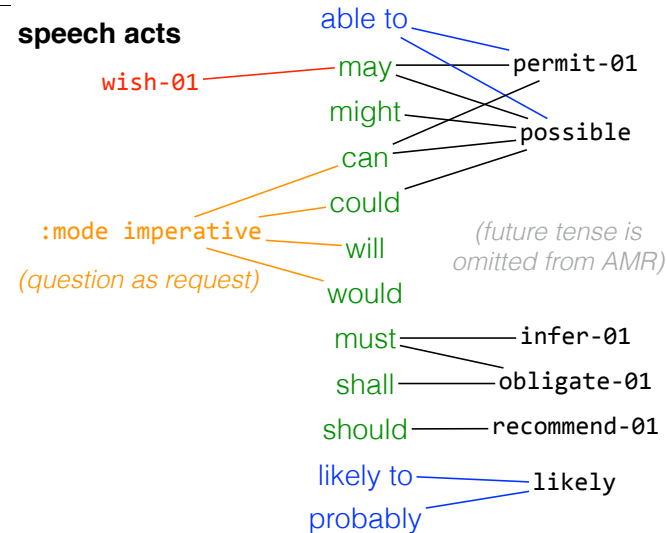
```
(p / permit-01)
:arg1 (I / leave)
:arg0 (y / you)
```

Example 2:

I *can* see Russia from my house!
I’m *able* to see Russia from my house!

```
(p1/1possible
:domain1(s / see-01
:arg0 (i / )
:arg1 (c1/1country1:name1(n1/1name1:op11"Russia"))
:location1(h1/1house1:poss1i)))
```

Speech Acts



Sentence Types

The most common patterns:

Types	AMR notation
indicative (declarative)	<i>(default)</i>
imperative (command)	:mode imperative <i>(with you arg if implied subject)</i>
interjection	:mode expressive
yes-no question	:mode interrogative
WH-question	amr)unknown
quotation without speech verb	(s1/1say)011:arg01<speaker>1...
vocative	(s1/1say)011:arg21<addressee>1...
polite ("please", etc.)	:polite1+

Questions ...

- **Question YES-NO** : « *Are you worried?* »
(w / worry-01
arg0 (y / you)
:mode interrogative)
- **Question WHY** : « *Why worry ? (what is the point of worrying ?)* »
(w / worry-01
arg0 (y / you)
:purpose (a / amr-unknown))
Think of amr)unknown as an in situ question pronoun. Structurally, the AMR is the same as a declarative sentence.
- **Question WHAT** : « *What 's the problem?* »
(p / problem
:domain (a / amr-unknown))
- **Question HOW MANY** : « *How many peppers did Peter Piper pick?* »
(p / pick-01
arg0 (p2 / person : name (n / name :op1 « Peter » :op2 « Piper »)))
arg1 (p3 / pepper
:quant (a / amr-unknown)))

Quantifications

Only explicit quantifiers are included in the AMR :

- « *Two apples* »
(a / apple-01
:quant 2)
- « *A lot of apples* »
(a / apple-01
:quant (1 / lot))
- « *Apples are fruit* »
(f / fruit
:domain (a / apple))
- « *All apples are fruit* »
(f / fruit
:domain (a / apple
:quant (a / all)))

Sets

Special predicate include-91 for explicitly mentioned sets :

- « *(I ate) 5 of the 12 donuts* » :
(d / donut : quant 5
arg1-of (i / include-91
:arg2 (d2 / donut :quant 12)))
- « *42% of the donuts* » :
(d / donut
arg1-of (i / include-91
:arg2 (d2 / donut)
:ARG3 (p / percentage-entity :value 42)))

Reification (1)

- « **The man at the store** » :

(m / man :location (s / store))

- « **The man *always* at the store** » :

- Need to “modify” the relation!
- Solution: Convert (“**reify**”) the relation w/ a special frame

(m / man
:arg1-of (b / **be-located-at-91**
:arg2 (s / store)
:time (a / **always**))

Reification (1)

Every role has a designed reification – either a verb frame or a special -91 frame :

- **have-purpose-91, have-polarity-91, have-part-91, ...**
- ~~have-topic-91~~ **concern-02**

« *These slides **are** about semantics* »

(c / **concern-02**
:arg0 (s / slide :mod (t / this))
:arg1 (s2 / semantics))

3. Conclusion

- Working with AMR Data
- AMR Strength & Limitations
- Further AMR documentation

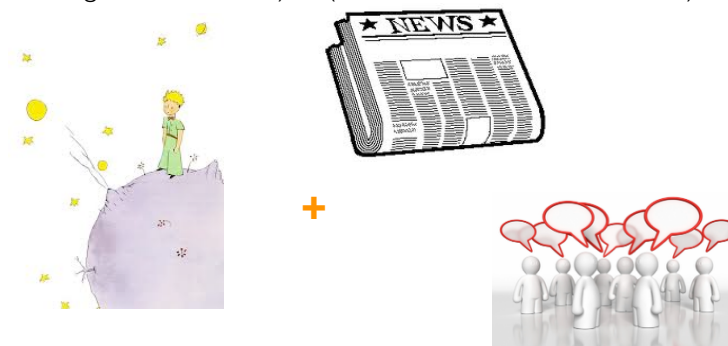
Datasets (1)

AMR Bank: *The Little Prince*

(novel—English translation)

LDC Releases

(news, discussion forums, ...)



= 15k AMRs
(with more to come!)

Datasets (2)

- Datasets: Details AMR Bank (Release 1.4; <http://amr.isi.edu/download/amr-bank-v1.4.txt>)
- English translation of *Le Petit Prince (The Little Prince)*, freely downloadable. 1,500 AMRs
- AMR Public Release 1.0 (LDC2014T12): largest public release w/ 13,051 AMRs
- DEFT Release 3 (LDC2013E117): evaluation data in Flanigan et al. 2014, Wang et al. 2015.
- DEFT Release 4 (LDC2014E41): largest release w/ 18,779 AMRs total
- DEFT Release 5 (Sep. 2015) will include wikification, (pretty much) no directed cycles
- Small (100-AMR) sets of Czech and Chinese AMRs have been annotated.
- Vanderwende et al. (2015) data to appear: several languages, automatically converted from logical forms
- PropBank will soon all be converted to AMR style (mapping nominalizations to verbs, etc.) and re-released.

Working with AMR Data

- AMRs are stored in a plain text format (in Penman notation)
- Script for loading them into a Python data structure: <https://github.com/nschneid/amr-hackathon/tree/master/src>
- Also accepts **aligned AMRs** from the ISI aligner

AMR Strength

- **Abstracting away from morphological & syntactic variability**
- Predicate-argument structures
 - *Core + non-core roles*
- Named entities & values
- Coreference (w/in sentence)
- Modality

AMR Limitations

- no “deep” **lexical semantics**
*Example: fruit/berry, buy/sell, kill/die are **formally unrelated***
- no deep treatment of **quantification & scope**
- (almost) no **information structure**
- nothing across sentences in a **discourse**...yet

Further AMR Documentation

- **Homepage:** <http://amr.isi.edu/>
- **Guidelines:** <https://github.com/amrisi/amr-guidelines/blob/master/amr.md>
- **Editor help pages**, especially AMR Dictionary
 - (<http://www.isi.edu/~ulf/amr/lib/amr-dict.html>)
- **Editor release search** (*rs query*) to check existing AMRs for a precedent