AMR Annotation:
Special Topics

http://tiny.cc/amrtutorial

Congratulations!

You now know how to AMR **simple sentences**.

BUT: English is a wily opponent.

- Copulas, light verbs

- Prepositions

- Derivational morphology

- Relational nouns

- Coordination & clausal connectives

- Modality

- Non-declarative sentences

- Questions

- Comparisons

- Quantification

- 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** sifts thistles*

- **Derivational morphology**
  - ‣ **Participles:** *thistle-sift**ing** person*
  - ‣ **Nominalizations:** *thistle sift**er***

    Concept may be implicit

- (p / person
    :ARG0-of (s / sift-01
            :ARG1 (t / thistle)))

Compositionality criterion

- We only "decompose" **derivational morphology** if a relative clause paraphrase is possible:

  ‣ teach**er** = person who teaches

    (p / person :ARG0-of (t / teach-01))

  ‣ profess**or** ≠ person who professes

    (p / professor)

Compositionality criterion

- Often core roles are available for modifiers:

  ‣ math teach**er** / teach**er** of math
  $\qquad\qquad$ = person who teaches math

  (p / person :ARG0-of (t / teach-01
  $\qquad\qquad\qquad\qquad\qquad$:ARG1 (m / math)))

  ‣ math profess**or** ≠ person who professes math

  (p / professor :mod (m / math))

8

Compositionality criterion

- Sometimes it is difficult to draw a line, but we do our best:

  ‣ opin**ion** = thing that is opined

    (t / thing :ARG1-of (o / opine-01))

  ‣ profess**ion** ≠ 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

I am your father

(h / **have-org-role-91**
   :ARG0 (h2 / he)
   :ARG1 (c / company
    :name (n / name :op1 "TWA"))
   :ARG2 (p / pilot))

(h / **have-rel-role-91**
   :ARG0 (i / i)
   :ARG1 (y / you)
   :ARG2 (f / father))

Coordination & Clausal Connectives

Example connectives	AMR treatment
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          (o / or           (c / contrast-01
   :op1 X            :op1 X             :ARG1 X
   :op2 Y            :op2 Y             :ARG2 Y)
   :op3 Z)          :op3 Z)
```

- "and", "or" take 2 or more conjuncts in sequence as `:op#`

Coordination

Coordination

- Rachael Ray finds inspiration in cooking her family and her dog.

```
(i / inspire-01
   :ARG0 (c / cook-01



                        )
   :ARG1 (p / person :name (n / name :op1 "Rachael" :op2 "Ray")))
```

Coordination

- Rachael Ray finds inspiration in cooking her family and her dog.

```
(i / inspire-01
   :ARG0 (c / cook-01
          :ARG0 p
          :ARG1 (a / and
                 :op1 (f / family


                                                     )
                 :op2 (d / dog          )))
   :ARG1 (p / person :name (n / name :op1 "Rachael" :op2 "Ray")))
```

Coordination

- Rachael Ray finds inspiration in cooking her family and her dog.

```
(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

- Rachael Ray finds inspiration in cooking her family and her dog.

```
(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

- Rachael Ray finds inspiration in cooking, her family, and her dog.

```
(i / inspire-01
   :ARG0 (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

- 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 non-core args

- Yesterday 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))
   :time (y / yesterday))
```

Coordination: copied predicates

- 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)))
```
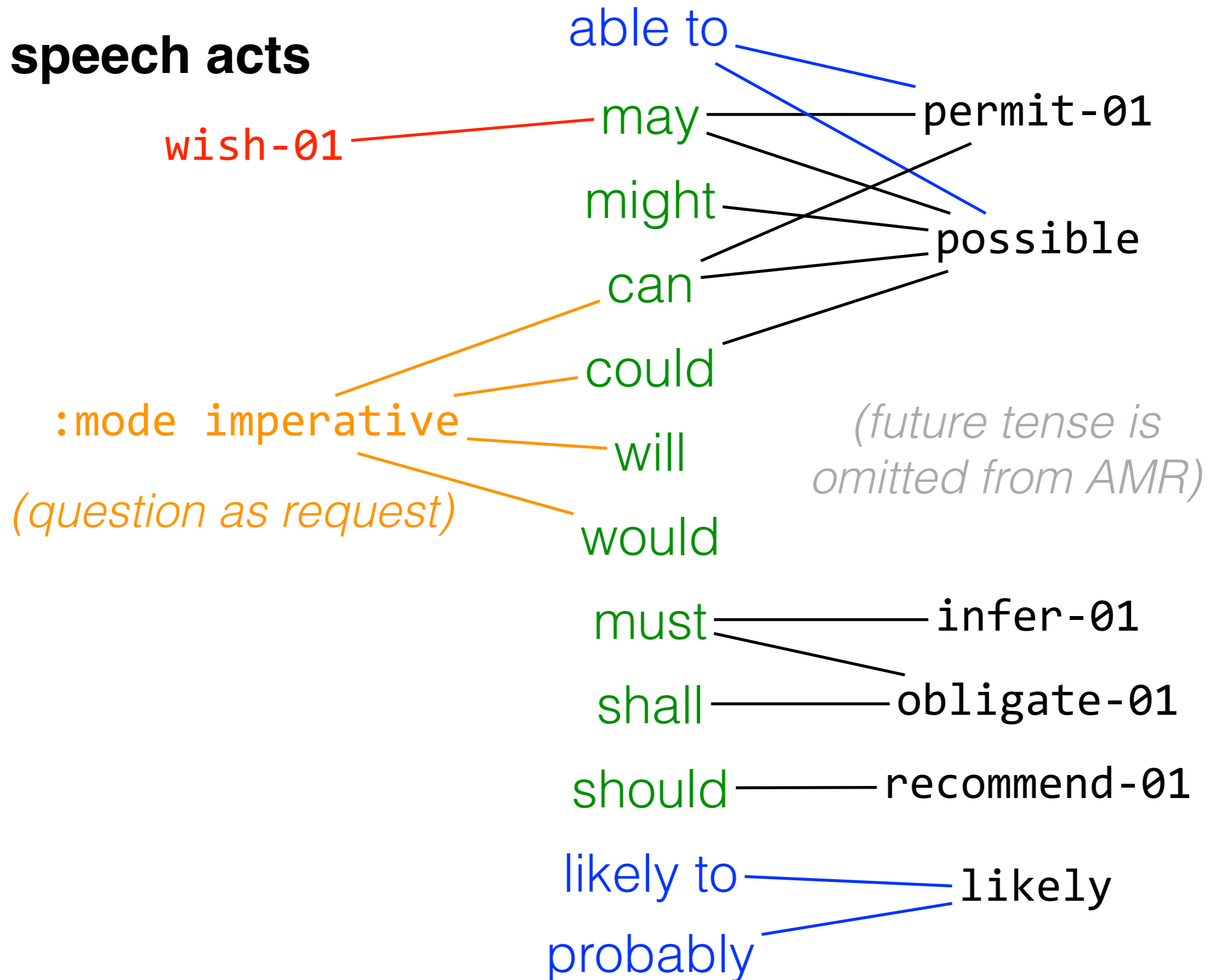
Modal Concepts

- You can leave.
  You may leave.
  It's all right for you to leave.

  ```
  (p / permit-01
     :ARG1 (l / leave-01
               :ARG0 (y / you))
  ```

- I can see Russia from my house!
  I'm able to see Russia from my house!

  ```
  (p / possible
     :domain (s / see-01
                 :ARG0 (i / i)
                 :ARG1 (c / country :name (n / name :op1 "Russia"))
                 :location (h / house :poss i)))
  ```

23

speech acts

able to

wish-01 —— may —— permit-01

might

possible

can

could

:mode imperative

(question as request)

will

(future tense is omitted from AMR)

would

must —— infer-01

shall —— obligate-01

should —— recommend-01

likely to —— likely

probably

24

Sentence Types

Type	AMR treatment
indicative (declarative)	*(default)*
imperative (command)	`:mode imperative` (with **you** arg if implied subject)
interjection	`:mode expressive`
yes-no question	`:mode interrogative`
WH-question	`amr-unknown`
quotation without speech verb	`(s / say-01 :ARG0 <speaker> …)`
vocative	`(s / say-01 :ARG2 <addressee> …)`
polite ("please", etc.)	`:polite +`

Questions: yes-no

- Are you worried?

```
(w / worry-01
   :ARG0 (y / you)
   :mode interrogative)
```

Questions: wh

- 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.

- What's the problem?

```
(p / problem
   :domain (a / amr-unknown))
```

- How many peppers did Peter Piper pick?

```
(p / pick-10
   :ARG0 (p2 / person :name (n / name :op1 "Peter" :op2 "Piper"))
   :ARG1 (p3 / pepper
              :quant (a / amr-unknown)))
```

27

Comparison

- The treatment of comparative constructions is changing.

- Apples are redder than bananas.

old way:

```
(r / red
   :domain (a / apple)
   :degree (m / more)
   :compared-to (b / banana))
```

new way:

```
(r / red
   :ARG1-of (h / have-degree-91
           :ARG2 (a / apple)
           :ARG3 (m / more)
           :ARG4 (b / banana)))
```

Quantification

- two apples

```
(a / apple
   :quant 2)
```

- a lot of apples

```
(a / apple
   :quant (l / lot))
```

Only explicit quantifiers are included in the AMR.

- All apples are fruit.

```
(f / fruit
   :domain (a / apple
              :quant (a / all)))
```

- Apples are fruit.

```
(f / fruit
   :domain (a / apple))
```

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

- *the man at the store*

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

- What about: *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)))

31

Reification

- Reification also allows a relational predicate to be focused.

- The man **is** at the store.

  (b / be-located-at-91
     :ARG1 (m / man)
     :ARG2 (s / store))

- I think the man **is** at the store.

  (t / think-01
     :ARG0 (i / i)
     :ARG1 (b / be-located-at-91
                :ARG1 (m / man)
                :ARG2 (s / store)))

Reification

- Every role has a designated 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))

English is a wily opponent.

- Copulas, light verbs

- Prepositions

- Derivational morphology

- Relational nouns

- Coordination & clausal connectives

- Modality

- Non-declarative sentences

- Questions

- Comparisons

- Quantification

- Subsets

- Reification

Other Phenomena

- Many other patterns for specific phenomena are documented in the **AMR Dictionary**. E.g.:

  ‣ *We'll eat **like** kings* → resemble-01

  ‣ *(Banarescu et al., 2013)* → publication-91

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

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

AMR Data

Real Data

- Thus far: mostly made-up examples

- Real sentences tend to be longer, but AMRed using the same principles

Real Data

- We need to borrow 55% of the hammer price until we can get planning permission for restoration which will allow us to get a mortgage.

Real Data

control structure

- We need to borrow 55% of the hammer price until we can get planning permission for restoration which will allow us to get a mortgage.

Real Data

subset → include-91

- We need to borrow 55% of the hammer price until we can get planning permission for restoration which will allow us to get a mortgage.

Real Data

temporal connective

- We need to borrow 55% of the hammer price until we can get planning permission for restoration which will allow us to get a mortgage.

43

Real Data

- We need to borrow 55% of the hammer price until we can get planning permission for restoration which will allow us to get a mortgage.

  modal

44

Real Data

- We need to borrow 55% of the hammer price until we can get planning permission for restoration which will allow us to get a mortgage.

deverbals → event predicates

Real Data

We need to borrow 55% of the hammer price until we can get planning permission for restoration which will allow us to get a mortgage.

```
(n / need-01
    :ARG0 (w / we)
    :ARG1 (b / borrow-01
        :ARG0 w



                            )



                            )
```

Real Data

We need to borrow 55% of the hammer price until we can get planning permission for restoration which will allow us to get a mortgage.

```
(n / need-01
    :ARG0 (w / we)
    :ARG1 (b / borrow-01
        :ARG0 w
        :ARG1 (p / percentage-entity :value 55
            :ARG1-of (i / include-91
                :ARG2 (p2 / price
                    :mod (h / hammer))))



                    )
```

Real Data

We need to borrow 55% of the hammer price until we can get planning permission for restoration which will allow us to get a mortgage.

```
(n / need-01
    :ARG0 (w / we)
    :ARG1 (b / borrow-01
        :ARG0 w
        :ARG1 (p / percentage-entity :value 55
            :ARG1-of (i / include-91
                :ARG2 (p2 / price
                    :mod (h / hammer))))
        :time (u / until
            :op1 (p3 / possible
                :domain (g / get-01
                    :ARG0 w
                    :ARG1 (p4 / permit-01
                        :ARG1 (p5 / plan-01)
                        :purpose-of (r / restore-01)
                        :ARG0-of (a / allow-01
                            :ARG1 (m / mortgage-01
                                :ARG0 w))))))))
```
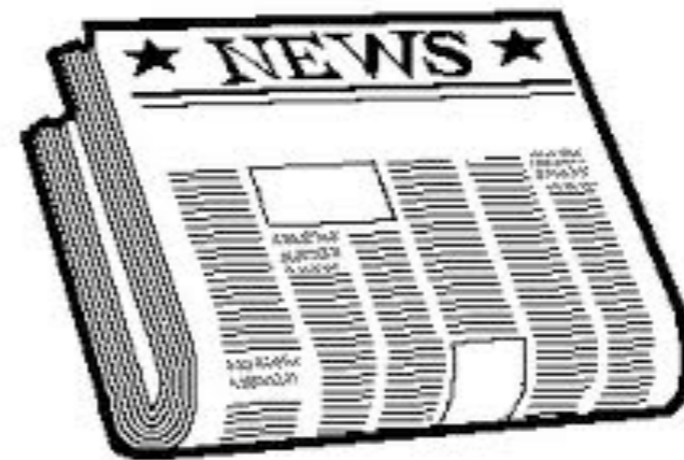
Datasets

AMR Bank: *The Little Prince*
(novel—English translation)

LDC Releases
(news, discussion forums, …)

+

= 15k AMRs
(with more to come!)

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
vs.
Other Formalisms

AMR Strengths

- **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**

  ‣ *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

Design Decisions

- AMR annotations are **not tied to individual words or any syntactic derivation**

- **Practicality for human annotators** is primary

  ‣ AMR makes no compromises for (current) algorithms

- **Single structure** rather than many layers

- Extensive **documentation and tool support**

Detailed Comparison

- See "Other Formalisms" slides