Meaning representation for natural language understanding

Mariana Romanyshyn
Grammarly, Inc.

Contents

1. Intro to meaning representation

How do we understand the meaning of a sentence

- we know what words mean
- we know how words interconnect
- we know the context of the sentence
- we know a lot about the world around us


PLEASE, HAVE A SEAT.


THANK YOU.
DOMICS

How do we understand the meaning of a sentence

- we know what words mean
- we know how words interconnect
- we know the context of the sentence
- we know a lot about the world around us
- ~~we use non-verbal communication~~

Can we teach the computer to understand the meaning of a sentence in the same way as humans do?

Option 1: bag of words

Text: *Children wanted to eat all cookies. They were delicious!*

Option 1: bag of words

Text: *Children wanted to eat all cookies. They were delicious!*

Meaning: *cookie, child, eat, delicious*

Option 1: bag of words

Text: *Cookies wanted to eat all children. They were delicious!*

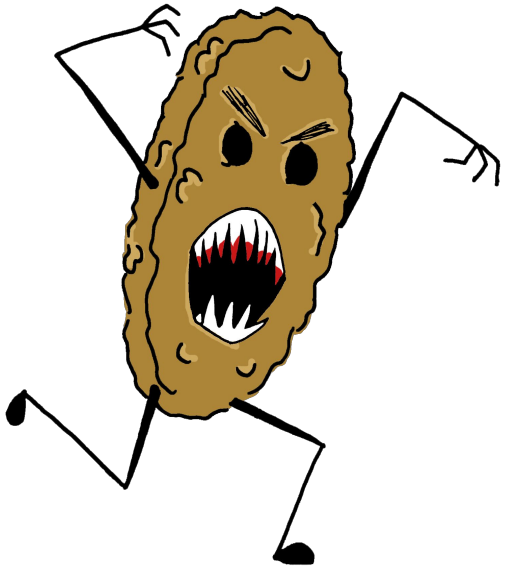


Image from https://www.shutterstock.com

Option 1: bag of words

Text 1: *Children wanted to eat all cookies. They were delicious!*

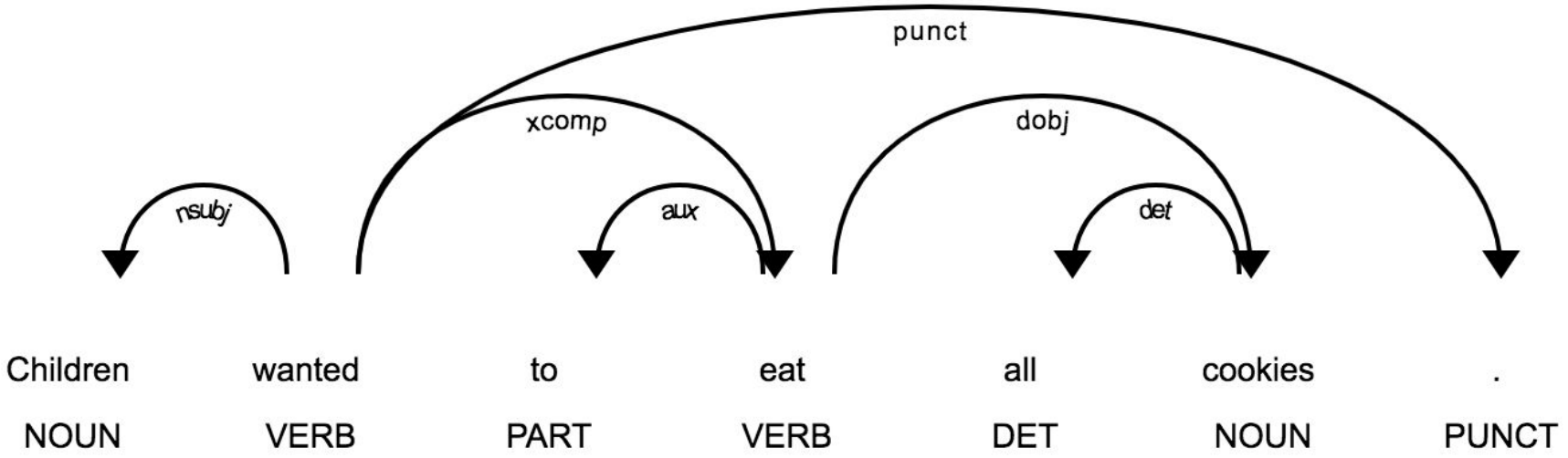
Meaning: *cookie, child, eat, delicious*

Text 2: *Cookies wanted to eat all children. They were delicious!*

Meaning: *cookie, child, eat, delicious*


I THINK WE CAN DO BETTER THAN THAT
makeameme.org

Option 2: syntactic parsing

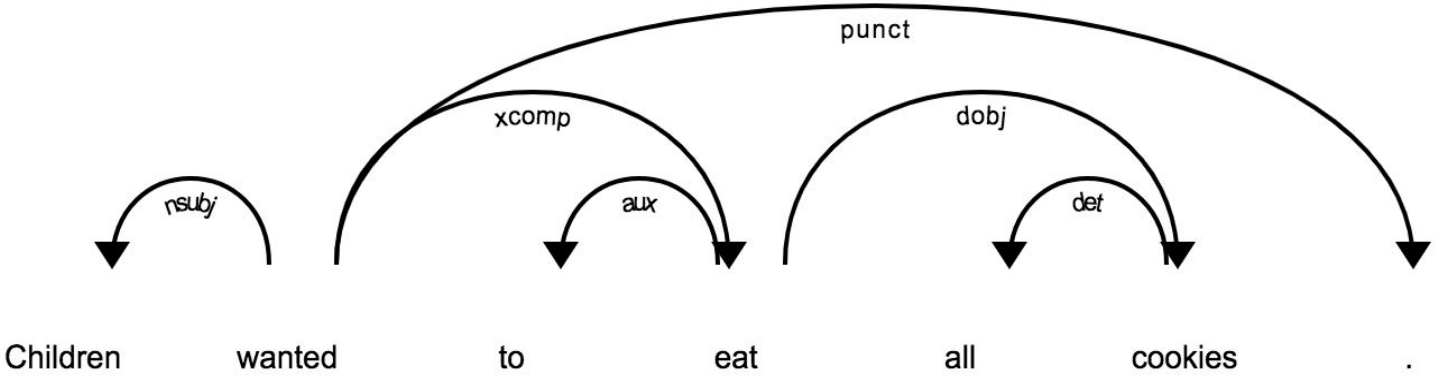


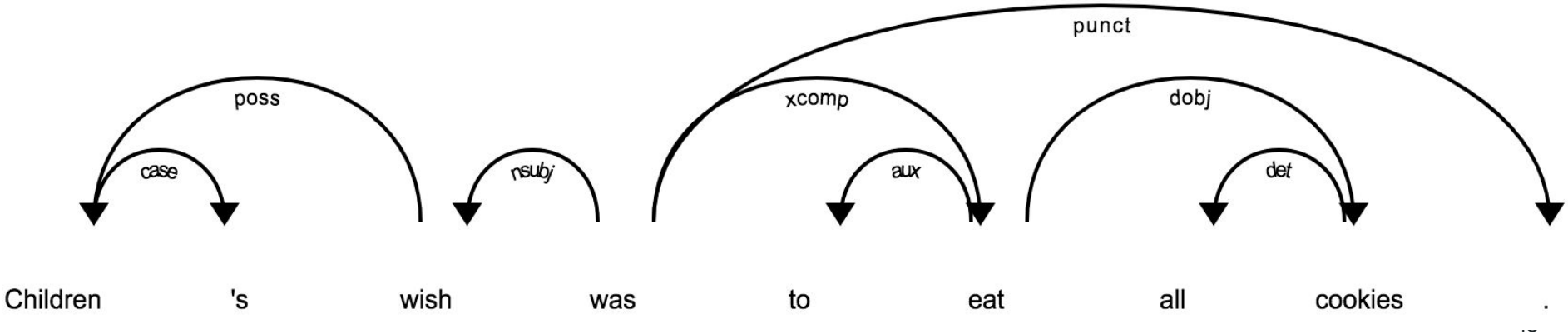
Option 2: syntactic parsing

Reality:

- the ***doer*** of the action is ***not*** always the ***subject***
- the ***action*** is ***not*** always presented by a ***verb***
- ***paraphrases*** should (ideally) have the ***same*** representation

Option 2: syntactic parsing






I THINK WE CAN DO BETTER THAN THAT
makeameme.org

Option 3: semantic parsing

Meaning representation formalisms:

- universal conceptual cognitive annotation
- typed lambda-calculus expressions
- minimal recursion semantics
- semantic role labelling
- abstract meaning representation (AMR)
- *… (a hundred more formalisms)*

Option 3: semantic parsing

Meaning representation formalisms:

- universal conceptual cognitive annotatio
- typed lambda-calculus expressions
- minimal recursion semantics
- semantic role labelling
- abstract meaning representation (AMR)
- *… (a hundred more formalisms)*

Lambda calculus

Idea: sentence ➔ a logical expression

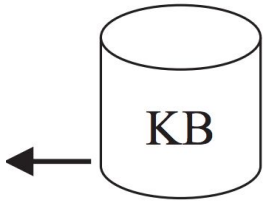
Children wanted to eat all cookies.

$$\exists x.child(x) \wedge \forall y.cookie(y) \wedge want(x, eat(x, y))$$

Lambda calculus

Sentence: *What is the capital of Germany?*
Which city is the capital of Germany?
Which city in Germany is the capital?

Logical form: `λx.capital(x, Germany)`



Result: *{Berlin}*

Semantic role labelling

Idea: detect who did what to whom

The police officer *detained* *the suspect* *at the crime scene*.

detainer **detainee** **location**

The suspect *was* *detained* *at the crime scene* *by* *the police officer*.

detainee **location** **detainer**

This is *the police officer* *who* *detained* *the suspect* *at the crime scene*.

detainer **detainee** **location**

Semantic role labelling: PropBank

PropBank frame for **“want”**:

- ARG0: wanter
- ARG1: thing wanted
- ARG2: beneficiary

PropBank frame for **“eat”**:

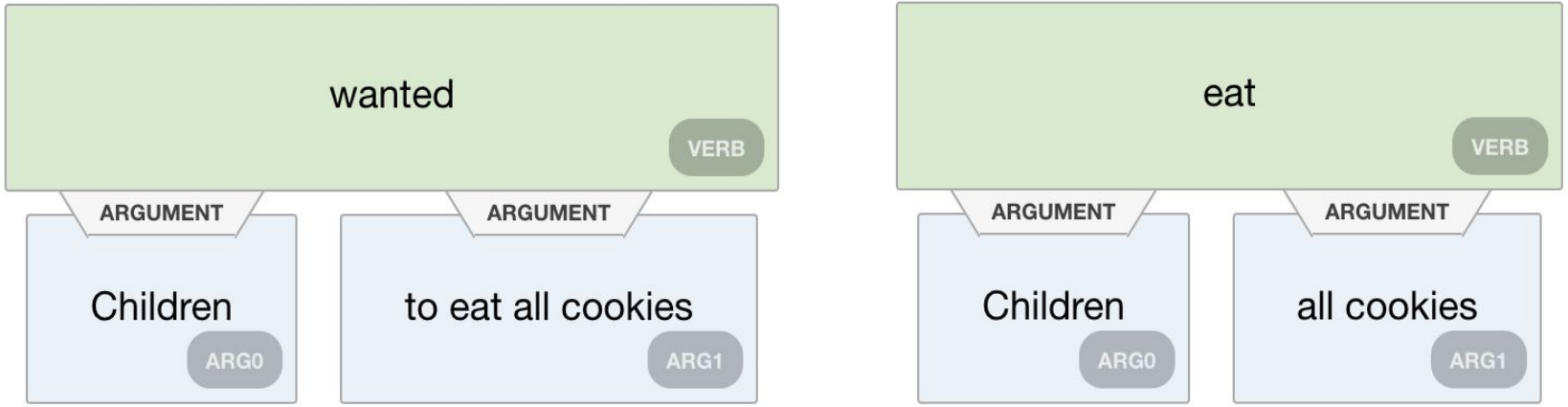
- ARG0: consumer, eater
- ARG1: meal

Additional roles:

- TMP: when?
- LOC: where?
- DIR: where to/from?
- MNR: how?
- CAU: why?
- etc.

Semantic role labelling: AllenNLP

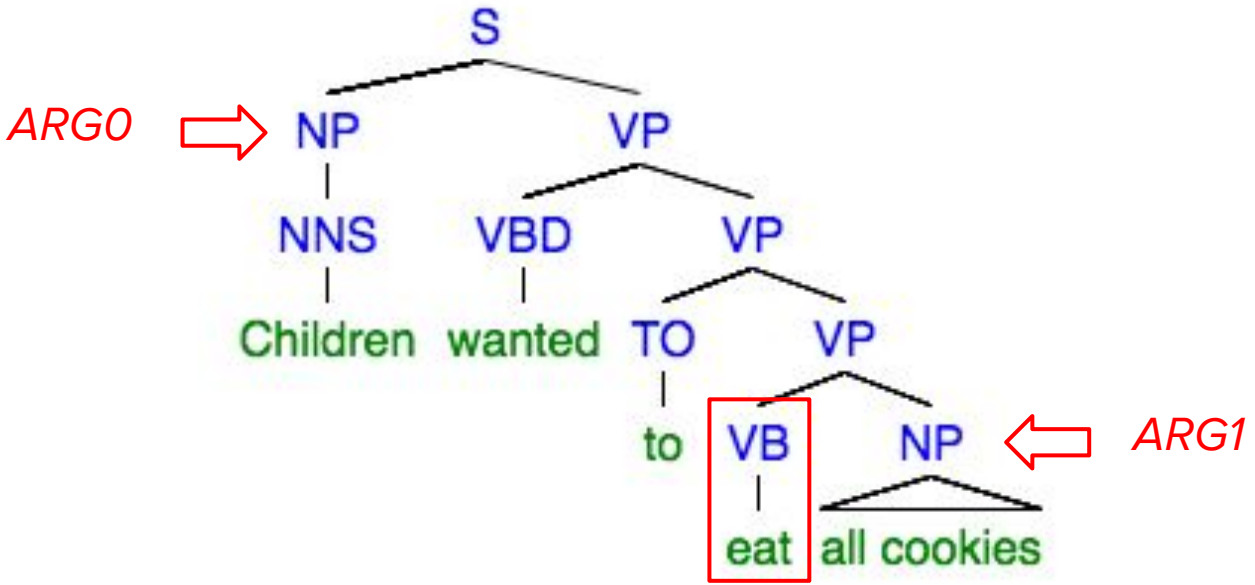
Children wanted to eat all cookies.



Built with https://demo.allennlp.org

Semantic role labelling

eat-01: [ARG0: Children] wanted to [V: eat] [ARG1: all cookies] .



Abstract meaning representation

Idea: abstract away from words to concepts

```
(w / want-01
   :ARG0 (c / child)
   :ARG1 (e / eat-01
            :ARG0 c
            :ARG1 (c2 / cookie
                      :mod (a / all))))
```

Children wanted to eat all cookies.
Children's wish was to eat all cookies.
Eating all cookies was what children desired.
Children felt like eating all cookies.

2. Exploring AMRs

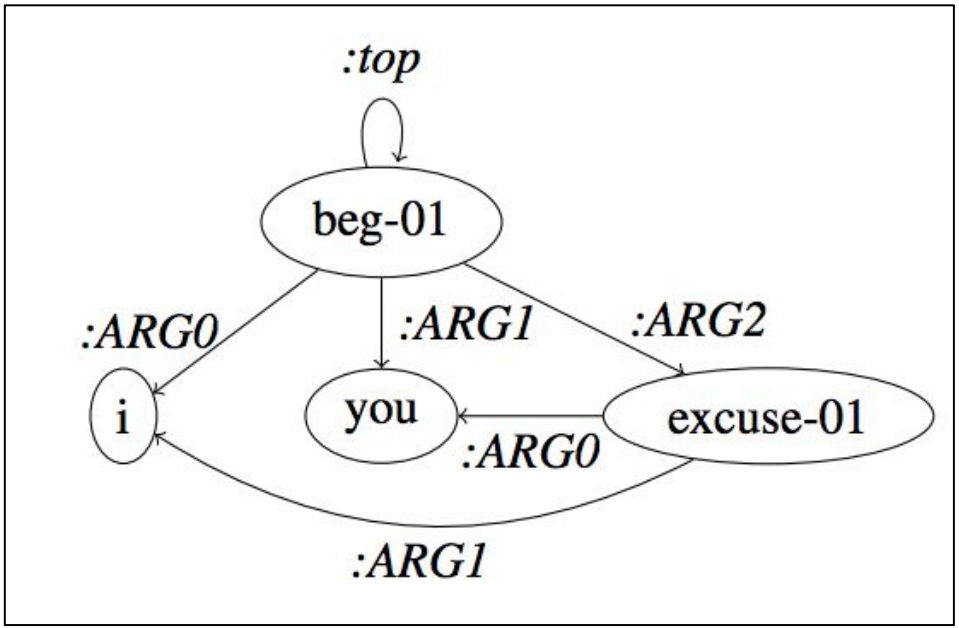
What is AMR?

Each sentence is a *rooted, directed, acyclic* graph, where:

- nodes are concepts
- edges are semantic relations
- function words are omitted

***I** beg you to excuse **me**.*

same node



AMR nodes

AMR nodes are concepts that can be:

- a PropBank frame (“beg-01”)
- a word (“you”, “boy”)
- a special keyword (“person”, “organization”, “date-entity”, “volume-quantity”, “temporal-quantity”, etc.)

AMR pros and cons

AMR handles:

- semantic roles
- entity types
- coreference
- modality
- polarity
- wikification

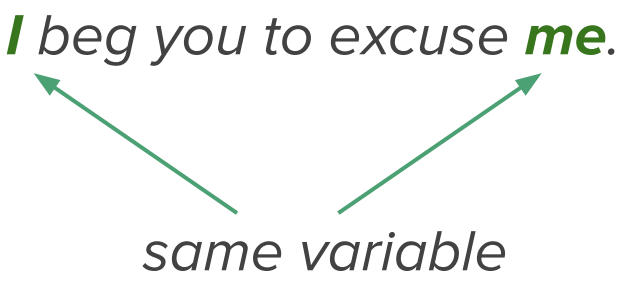
AMR doesn’t handle:

- tense
- definiteness
- plurality

AMR notation

AMR uses PENMAN notation, where each node is assigned a variable:

```
(b / beg-01
    :ARG0 (i / i)
    :ARG1 (y / you)
    :ARG2 (e / excuse-01
        :ARG0 y
        :ARG1 i))
```



AMR examples: frames

The boy desires the girl to believe him.
The boy wants the girl to believe him.
The boy desires to be believed by the girl.
The boy has a desire to be believed by the girl.
The boy's desire is for the girl to believe him.
The boy is desirous of the girl believing him.

```
(w / desire-01
   :ARG0 (b / boy)
   :ARG1 (b2 / believe-01
             :ARG0 (g / girl)
             :ARG1 b))
```

AMR examples: polarity

The soldier was not afraid of dying.
The soldier was not afraid to die.
The soldier did not fear death.

```
(f / fear-01
      :polarity "-"
      :ARG0 (s / soldier)
      :ARG1 (d / die-01
                  :ARG1 s))
```

AMR examples: modality

The boy must not go.
It is obligatory that the boy not go.

```
(o / obligate-01
   :ARG2 (g / go-02
            :ARG0 (b / boy)
            :polarity -))
```

AMR examples: unknown variables

Which state borders with Kansas?

```
(b / border-01
   :ARG0 (s / state
            :name (n / name :op1 (a / amr-unknown)))
   :ARG1 (s2 / state
            :name (n2 / name :op1 “Kansas”)))
```

AMR examples: unknown variables

Does Texas border with Kansas?

```
(b / border-01
   :ARG0 (s / state
            :name (n / name :op1 “Texas”))
   :ARG1 (s2 / state
            :name (n2 / name :op1 “Kansas”))
   :polarity (a / amr-unknown))
```

14,000 people fled their homes at the weekend after a tsunami warning was issued, the UN said on its web site.

```
(s / say-01
  :ARG0 (o / organization
           :name (n / name :op1 "UN"))
  :ARG1 (f / flee-01
           :ARG0 (p / person :quant 14000)
           :ARG1 (h / home :poss p)
           :time (w / weekend)
           :time (a2 / after
                    :op1 (w2 / warn-01
                             :ARG1 (t / tsunami))))
  :medium (s2 / site
             :poss o
             :mod (w3 / web)))
```

3. Building AMRs

AMR parsing

Data - AMR Banks in PENMAN format:

- 1.K sentences from *The Little Prince* :)
- 40K sentences of newswire, discussion forum and other web logs, television transcripts
- 7K sentences of Bio AMR (cancer-related PubMed articles)

AMR parsing

You become responsible , forever , for what you have tamed .

```
(b / become-01
      :ARG1 (y / you)
      :ARG2 (r / responsible-03
            :ARG0 y
            :ARG1 (t2 / thing
                  :ARG1-of (t / tame-01
                        :ARG0 y))
            :extent (f / forever)))
```

AMR parsing

Parsing algorithms:

- graph-based
- transition-based
- ~~rule-based~~
- ~~seq2seq-based~~

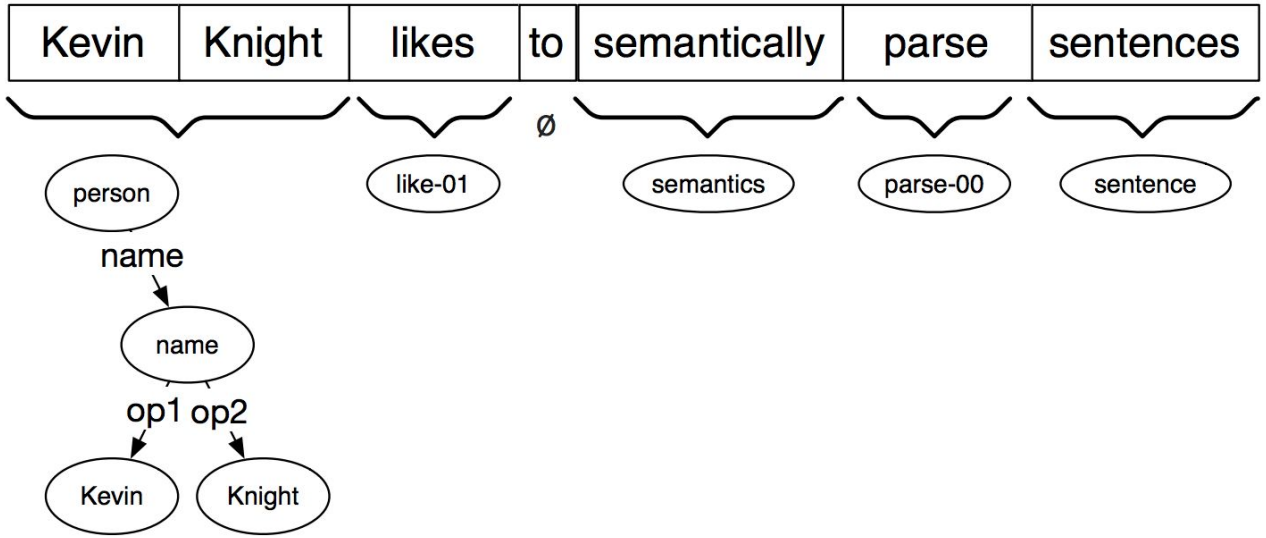
What is needed:

- POS tagging
- named-entity recognition
- syntactic parsing
- coreference resolution

Graph-based parsing

JAMR parser:

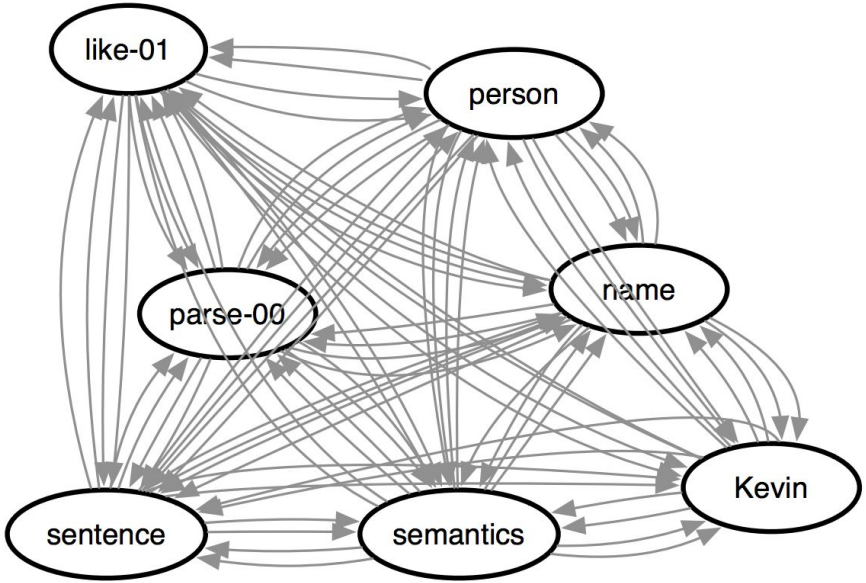
- identify concepts
- identify relations between concepts



Graph-based parsing

JAMR parser:

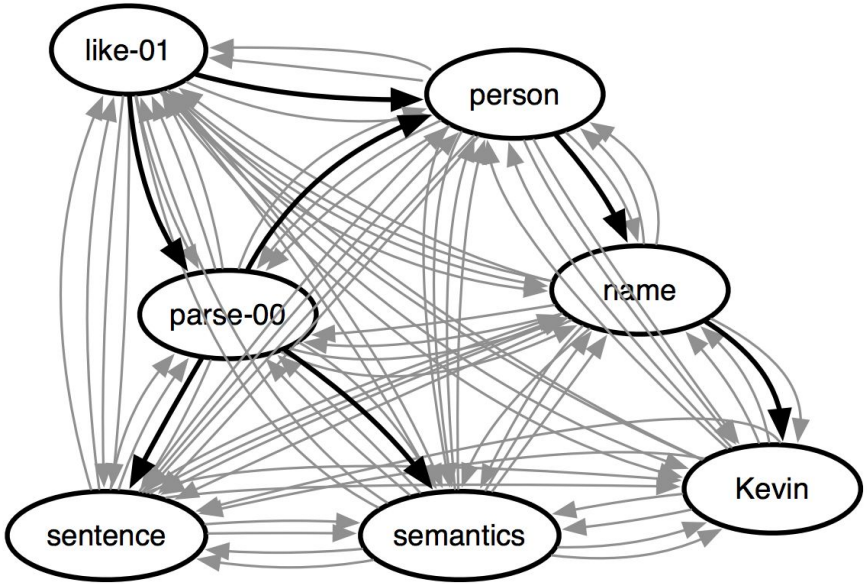
- identify concepts
- identify relations between concepts



Graph-based parsing

JAMR parser:

- identify concepts
- identify relations between concepts

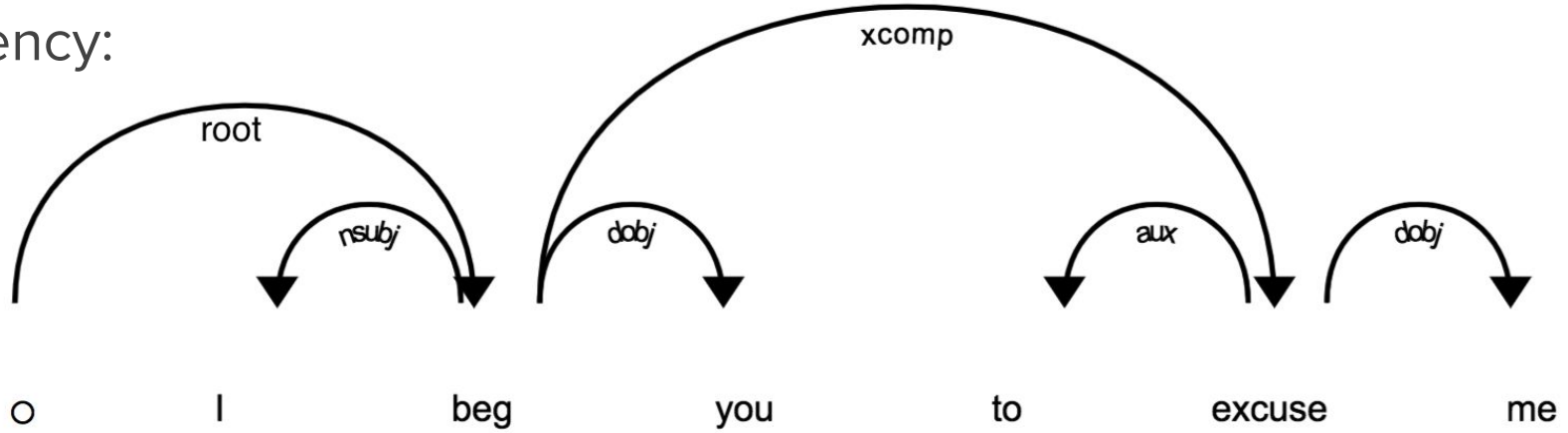


Transition-based parsing

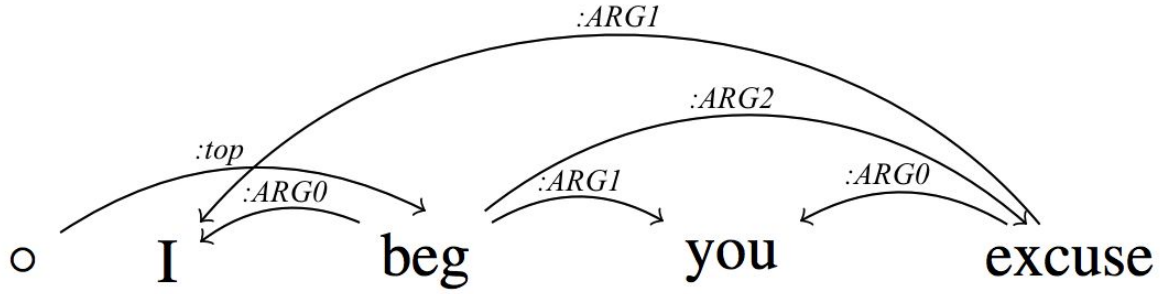
AMR graphs look similar to dependency parse trees, don’t they?

Transition-based parsing

Dependency:



AMR:



Transition-based parsing

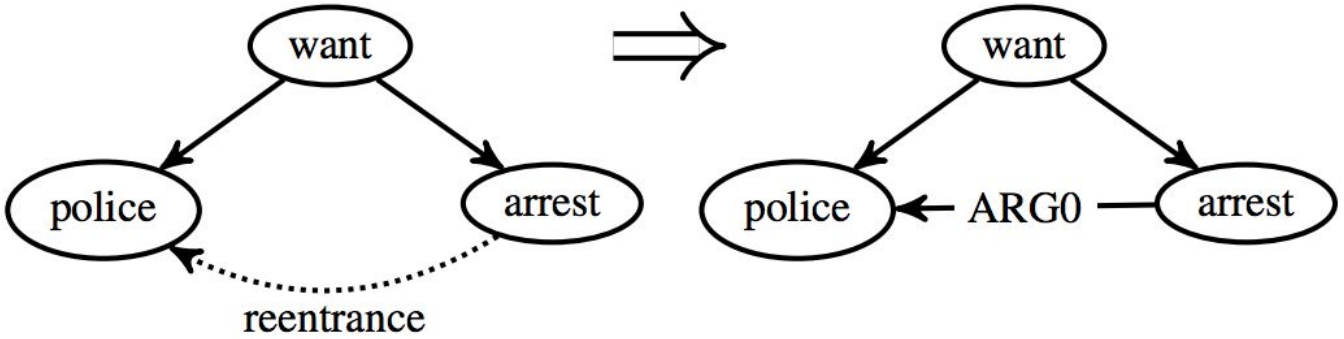
CAMR parser - transform a dependency parse tree into AMR

- traverse the dependency tree
- at each node/edge, collect features and classify action
  - merge nodes
  - swap nodes
  - delete a node
  - replace node
  - re-enter a node
  - attach edge
  - delete edge and re-attach to a new node
  - label with concept

Transition-based parsing

CAMR parser - transform a dependency parse tree into AMR

- traverse the dependency tree
- at each node/edge, collect features and classify action



AMR evaluation: smatch

```
(w / want-01
   :ARG0 (b / boy)
   :ARG1 (b2 / believe-01
             :ARG0 (g / girl)
             :ARG1 b))
```

```
instance(w, want-01)          /* w is an instance of wanting */
instance(b, boy)              /* b is an instance of boy */
instance(b2, believe-01)      /* b2 is an instance of believing */
instance(g, girl)             /* g is an instance of girl */
ARG0(w, b)                    /* b is the wanter in w */
ARG1(w, b2)                   /* b2 is the wantee in w */
ARG0(b2, g)                   /* g is the believer in b2 */
ARG1(b2, b)                   /* b is the believee in b2 */
```

AMR evaluation

- F1 smatch (up to 70%)
  - graph-based > transition-based

- Speed
  - graph-based < transition-based

4. Application of AMR

Application of AMR

- natural language generation (SemEval shared task, 2017)
- information extraction (Rao et al., 2017)
- entity linking (Pan et al., 2015)
- text summarization (Liu et al., 2015; Takase et al., 2016, Dohare et al., 2017, Liao et al., 2018)
- question answering (Jurczyk and Choi, 2015)
- machine comprehension (Sachan and Xing, 2016)

Natural Language Generation: example

```
(a / and
    :op1 (r / remain-01
            :ARG1 (c / country :wiki "Bosnia_and_Herzegovina"
                              :name (n / name :op1 "Bosnia"))
            :ARG3 (d / divide-02
                     :ARG1 c
                     :topic (e / ethnic)))
    :op2 (v / violence
            :time (m / match-03 :mod (f2 / football)
                     :ARG1-of (m2 / major-02))
            :location (h / here)
            :frequency (o / occasional))
    :time (f / follow-01
             :ARG2 (w / war-01
                      :time (d2 / date-interval
                              :op1 (d3 / date-entity :year 1992)
                              :op2 (d4 / date-entity :year 1995))))))
```

Natural Language Generation: example

Source	Text
Reference	following the 1992-1995 war bosnia remains ethnically divided and violence during major football matches occasionally occurs here.
RIGOTRIO	following the 1992 1995 war, bosnia has remained an ethnic divide, and the major football matches occasionally violence in here.
CMU	following the 1992 1995 war , bosnia remains divided in ethnic and the occasional football match in major violence in here

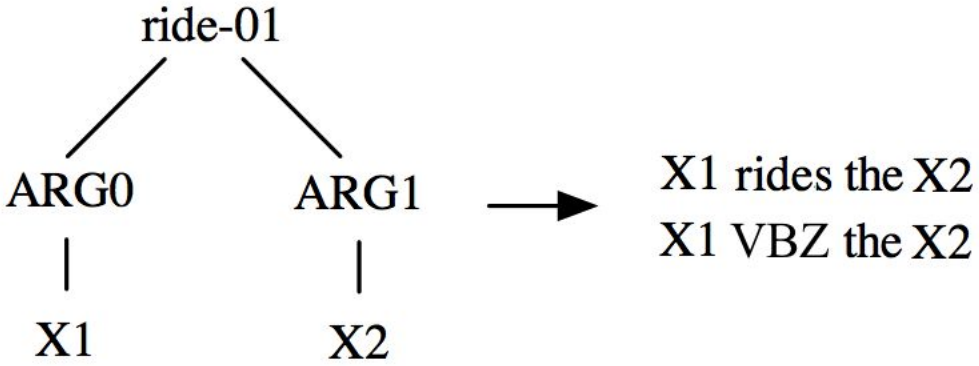
Example from SemEval 2017 shared task

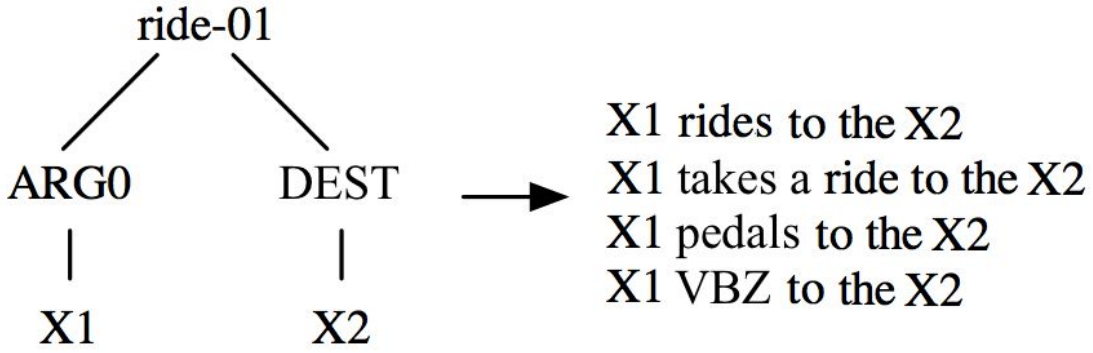
Natural Language Generation

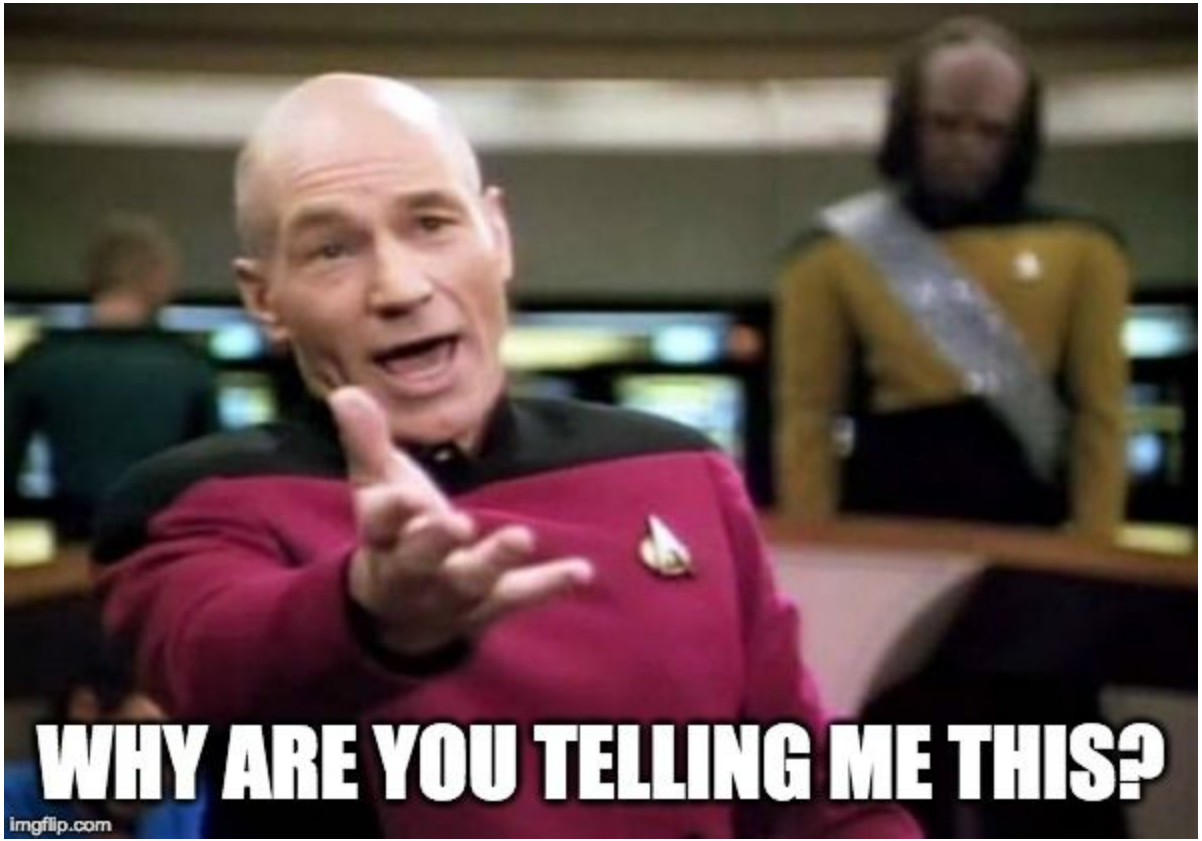
Idea:

- extract rules from AMR Banks
- use rules as features for a transducer
- find the best-scoring transduction (SMT-kind)
- pick the highest scoring sentence with a language model

Natural Language Generation






WHY ARE YOU TELLING ME THIS?
imgflip.com

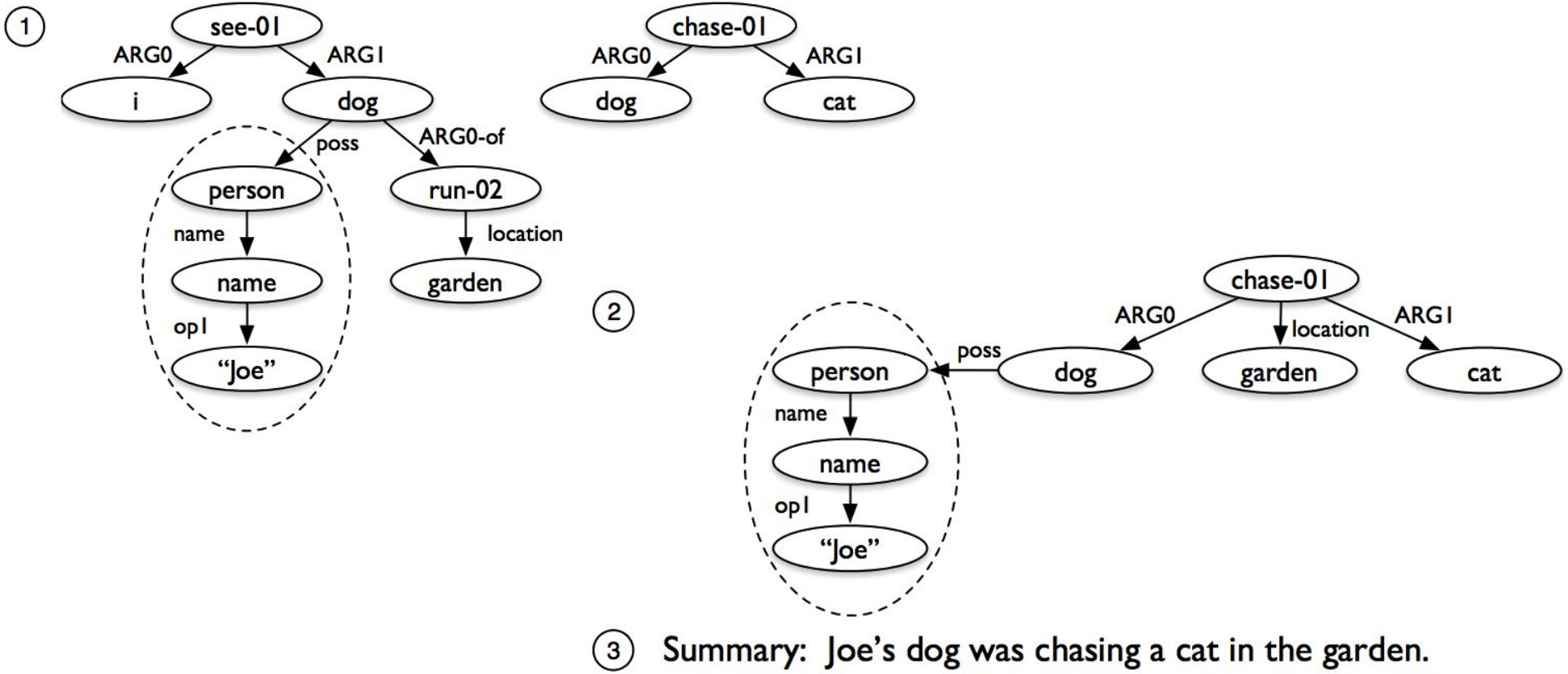
Abstractive text summarization

Idea:

- sentences in the document ➜ AMR graphs
- AMR graphs ➜ summary graph
- summary graph ➜ summary sentences

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

Sentence B: The dog was chasing a cat.

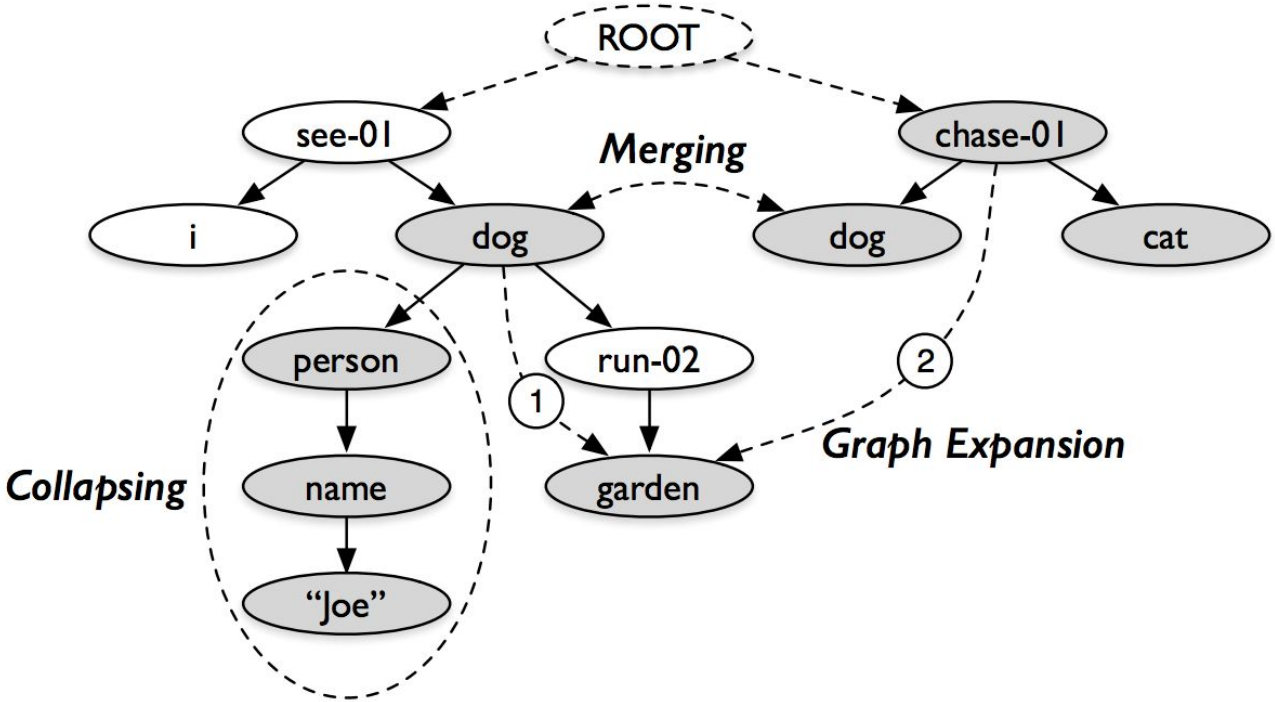


Example from Liu et al (2015)

Abstractive text summarization

How to build a summary graph:

- build a document-level AMR
- predict the summarizing subgraph



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

Sentence B: The dog was chasing a cat.

Example from Liu et al (2015)

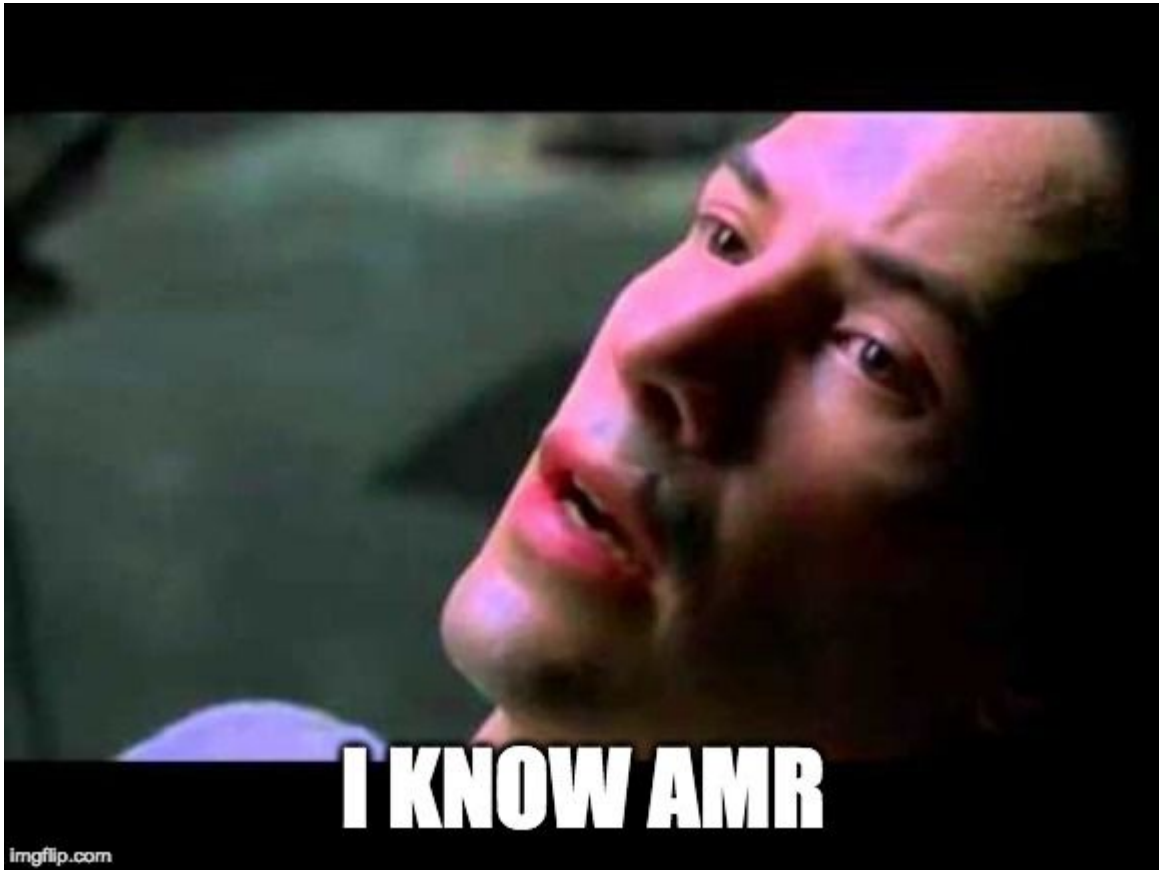
Abstractive text summarization

How to build a summary graph:

- build a document-level AMR
- predict the most meaningful subgraph
  - predict nodes
  - predict edges
  - ensure the result is a valid graph

Training data

- texts and their summaries parsed into AMRs

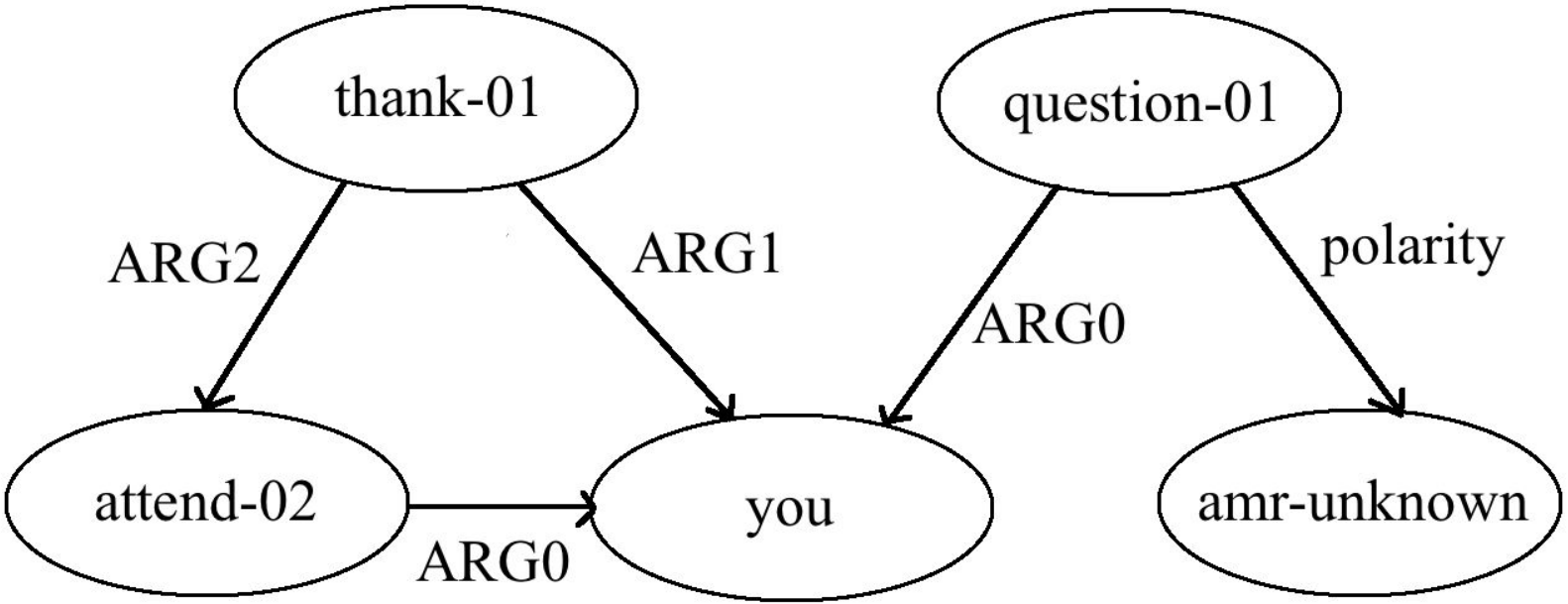

I KNOW AMR
imgflip.com

Useful references

- Building a Semantic Parser Overnight, Wang, Berant and Liang (2015)
- Abstract Meaning Representation for Sembanking, Banarescu et al. (2013)
- The Logic of AMR, Schneider et al. (2015)
- Graph-based AMR Parsing with Infinite Ramp Loss, Flanigan et al. (2016)
- A Transition-based Algorithm for AMR Parsing, Wang et al. (2015)
- An Incremental Parser for Abstract Meaning Representation, Damonte et al. (2017)
- Generation from Abstract Meaning Representation using Tree Transducers, Flanigan (2016)
- SemEval-2017 Task 9: Abstract Meaning Representation Parsing and Generation, May and Priyadarshi (2017)
- Neural AMR: Sequence-to-Sequence Models for Parsing and Generation, Konstas et al. (2017)

Useful references

- AMR Banks: https://amr.isi.edu/download.html
- Parser for the PENMAN format: https://github.com/goodmami/penman
- JAMR parser and generator: https://github.com/jflanigan/jamr
- AMREager parser: http://cohort.inf.ed.ac.uk/amreager.html
- CAMR parser: https://github.com/c-amr/camr
- Summarization with AMR: https://github.com/summarization/semantic_summ


thank-01
question-01
ARG2
ARG1
ARG0
polarity
attend-02
you
amr-unknown
ARG0