Abstract Meaning Representation

Computational Linguistics
Emory University
Jinho D. Choi
Abstract Meaning Representation

Introduced in 1998.

The dog will eat the bone.
The dog ate the bone.
A dog is eating a bone.

Revisited in 2013.

PropBank, named entities, coreferences, etc.
Pennman Notation

A simple way of representing directed graphs.

The dog is eating a bone.

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

Variables Relations Concepts

e/eat-01

ARG0 ARG1

d/dog b/bone
Reentrancy

The dog is eating a bone.

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

The dog wants to eat a bone.

(w / want-01
 :ARG0 (d / dog)
 :ARG1 (e / eat-01
 :ARG0 d
 :ARG1 (b / bone)))
Inverse Relations and Focus

The dog \textcolor{red}{ate}. Focus? The \textcolor{red}{dog that ate.}

(e / \textcolor{red}{eat-01} :ARG0 (d / \textcolor{red}{dog}))

(d / \textcolor{red}{dog} :ARG0-of (r / \textcolor{red}{ran-01}))

The dog ate the bone that it found.

X ARG0 Y \iff Y ARG0-of X

\begin{tikzpicture}
    \node (x) at (0,0) {e/eat-01};
    \node (y) at (1,0) {d/dog};
    \node (z) at (1,-1) {b/bone};
    \node (w) at (0,-1) {f/find-01};
    \draw [dotted] (x) -- (y);
    \draw [dotted] (x) -- (z);
    \draw [dotted] (x) -- (w);
    \draw [dotted] (y) -- (z);
    \draw [dotted] (y) -- (w);
    \draw [dotted] (z) -- (w);
\end{tikzpicture}
Inverse Relations and Focus

The dog ate the bone that it found.

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

Relations with no variable.

Negations

The dog did\textbf{ not} eat the bone.

\begin{verbatim}
(e /eat-01
  :polarity -
  :ARG0 (d / dog)
  :ARG1 (b / bone))
\end{verbatim}

Numbers

The dog ate \textbf{four} bones.

\begin{verbatim}
(e /eat-01
  :ARG0 (d / dog)
  :ARG1 (b / bone
    :quant 4))
\end{verbatim}
AMR Notation

The dog ate four bones that it found.

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

Relation :quant 4
:ARG1-of (f / find-01
:ARG0 (d))

Inverse Relation Reentrancy
AMR Framesets

Frames are generalized across similar concepts.

My fear of snakes
I am fearful of snakes
I fear snakes
I’m afraid of snakes

Frames are not generalized across synonyms.

I fear snakes
I’m terrified of snakes
Snakes creep me out

fear-01
teerify-01
creep-03
Stemming

All concepts drop plurality, articles, and tenses.

A dog  Eats
dogs  Eating
The dog  Ate
The dogs  Will eat
(d / dog)  (e / eat-01)

Consistent across cross-linguistically.
Abstraction

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.

(d / describe-01
 :ARG0 (m / man)
 :ARG1 (m2 / mission)
 :ARG2 (d / disaster))
Non-Core Role Inventory

Arguments that are not numbered.

:accompanyer, :age
:beneficiary
:compared-to, :concession, :condition, :consist-of
:degree, :destination, :direction, :domain, :duration
:example, :extent
:frequency
:instrument
:location
:manner, :medium, :mod, :mode
:name
:ord
:part, :path, :polarity, :poss, :purpose
:quant
:scale, :source, :subevent
:time, :topic, :unit
:value
:wiki
More Inventory

**Apples and Bananas**

(a / and
  :op1 (a2 / apple)
  :op2 (b / banana))

**Tuesday the 19th**

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

**Barack Obama**

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

**Five bucks**

(m / monetary-quantity
  :unit dollar
  :quant 5)
Dependency Tree vs. AMR Graph

*Jinho Choi wants to buy a car in Atlanta*

**Dependency Tree**

- **wants**
  - nsubj
  - xcomp

- **Choi**
  - nn

- **buy**
  - aux
  - dobj
  - prep

- **Jinho**
  - det
  - pobj

- **a**
  - det

- **Atlanta**
  - pobj

**AMR Graph**

- **want-01**
  - A0
  - A1

- **person**
  - op1
  - op2

- **car**
  - name

- **Atlanta**
  - name

- **Jinho**
  - A0

- **Choi**
  - A1

- **location**
Jinho Choi wants to buy a car in Atlanta

**AMR Graph**

- **want-01**
  - **person**
    - **name**
      - Jinho
      - Choi
  - **buy-01**
    - **car**
    - **Atlanta**

**Span Graph**

- **want-01[3:4]**
  - **person**
    - +name[1,3]
  - **buy-01[5:6]**
    - **[7:8]**
    - **[9:10]**

- **location**
  - **A1**
  - **A0**
Transition-based AMR Parsing

Populate $\sigma$ by post-traversing the dependency graph.

Populate $\beta$ with $\sigma$'s children.
Transition-based AMR Parsing

- **σ**
  - Jinho
  - Choi
  - to
  - a
  - car
  - Atlanta
  - in
  - buy
  - wants
  - root

- **β**

- **NEXT NODE-λc**
  \[(σ_0|σ_1|σ',|, G) \rightarrow (σ_1|σ', β = CH(σ_1, G'), G')\]

- **DELETE NODE**
  \[(σ_0|σ_1|σ',|, G) \rightarrow (σ_1|σ', β = CH(σ_1, G'), G')\]

- **γ[σ_0 \rightarrow λc]**

- **β = ∅**
Transition-based AMR Parsing

\[
\begin{array}{c|c}
\hline
\text{Jinho Choi} & \text{root} \\
\text{to} & \text{wants} \\
\text{a} & \text{Choi} \\
\text{car} & \text{buy} \\
\text{Atlanta} & \text{in} \\
\text{in} & \text{Jinho} \\
\text{buy} & \text{to} \\
\text{wants} & \text{car} \\
\text{root} & \text{in} \\
\hline
\end{array}
\]

\[
\sigma \\
\beta
\]

\[
(\sigma_0 | \sigma', \beta_0 | \beta', G) \Rightarrow (\tilde{\sigma} | \sigma', \beta', G')
\]
Transition-based AMR Parsing

<table>
<thead>
<tr>
<th>Jinho Choi</th>
<th>to</th>
<th>a</th>
<th>car</th>
<th>Atlanta</th>
<th>in</th>
<th>buy</th>
<th>wants</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

\( \sigma \)

\( \beta = \emptyset \)

**Table 1:** Transitions designed in our parser.

- **NEXT NODE-\(l_c\):**
  \[ (\sigma_0|\sigma_1|\sigma', [], G) \Rightarrow (\sigma_1|\sigma', \beta = CH(\sigma_1, G'), G') \]

- **DELETE NODE:**
  \[ (\sigma_0|\sigma_1|\sigma', [], G) \Rightarrow (\sigma_1|\sigma', \beta = CH(\sigma_1, G'), G') \]

\( \gamma[\sigma_0 \rightarrow l_c] \)

**Figure 4:** SWAP action

**Figure 5:** Shows an example where node (Washington) inherits all the incoming and outgoing arcs of node (Atlanta), and reattaches it with node (Washington). This action removes (Atlanta) and inserts it into the partial graph un-associated with (Atlanta).

Result state: \( \beta = \emptyset \)
Transition-based AMR Parsing

\[
(\sigma_0|\sigma', \beta_0|\beta', G) \Rightarrow (\beta_0|\sigma', \beta = \text{CH}(\beta_0, G'), G')
\]
Transition-based AMR Parsing

Table 1: Transitions designed in our parser.

<table>
<thead>
<tr>
<th>Transition</th>
<th>Precondition</th>
<th>Action</th>
<th>Result state</th>
</tr>
</thead>
<tbody>
<tr>
<td>NEXT NODE-$l_c$</td>
<td>$(\sigma_0</td>
<td>\sigma_1</td>
<td>\sigma', [], G)$</td>
</tr>
<tr>
<td>DELETE NODE</td>
<td>$(\sigma_0</td>
<td>\sigma_1</td>
<td>\sigma', [], G)$</td>
</tr>
</tbody>
</table>

...
Transition-based AMR Parsing

\[
(\sigma_0|\sigma', \beta_0|\beta', G) \Rightarrow (\sigma_0|\sigma', \beta', G') \mid \delta[(\sigma_0, \beta_0) \rightarrow l_r]
\]
Transition-based AMR Parsing

\[ \begin{array}{c|c|c|c}
\text{NEXT NODE-}l_c & \sigma_0|\sigma_1|\sigma', [], G \Rightarrow (\sigma_1|\sigma'|, \beta = CH(\sigma_1|\sigma', G'), G') & \gamma[\sigma_0 \rightarrow l_c] \\
\text{DELETE NODE} & (\sigma_0|\sigma_1|\sigma', [], G) \Rightarrow (\sigma_1|\sigma', \beta = CH(\sigma_1|\sigma', G'), G') & \text{NONE}
\end{array} \]
Transition-based AMR Parsing

\[ (\sigma_0|\sigma', \beta_0|\beta', G) \Rightarrow (\sigma_0|\sigma', \beta_0|\beta', G') \mid \delta[(k, \beta_0) \rightarrow l_r] \]

**Table 1:** Transitions designed in our parser.

South AMR graph. Figure 4 gives an example of ap-
Transition-based AMR Parsing

\[ \delta[(\sigma_0, \beta_0) \rightarrow l_r] \]

\( (\sigma_0|\sigma', \beta_0|\beta', G) \Rightarrow (\sigma_0|\sigma', \beta', G') \)

\( \sigma \)

\( \beta \)

\( \text{Jinho Choi} \)

\( \text{buy} \)

\( \text{person+name} \)

\( \text{wants} \)

\( \text{roots} \)

\( \text{car} \)

\( \text{Atlanta} \)

\( \text{Atlanta} \)

\( \text{location} \)

\( \text{car} \)

\( \text{Atlanta} \)

\( \text{location} \)

\( \text{car} \)
Transition-based AMR Parsing

Table 1: Transitions designed in our parser.

<table>
<thead>
<tr>
<th>Transition</th>
<th>State</th>
<th>Action</th>
<th>Result State</th>
</tr>
</thead>
<tbody>
<tr>
<td>NEXT NODE-( l_c )</td>
<td>( \sigma ) (</td>
<td>\sigma'</td>
<td>[], G )</td>
</tr>
<tr>
<td>DELETE NODE</td>
<td>( \sigma ) (</td>
<td>\sigma'</td>
<td>[], G )</td>
</tr>
</tbody>
</table>

\( \beta = \emptyset \)

Jinho Choi

person+name

buy

buy-01

wants

want-01

root

Atlantic

car

car

location

South

Israel

blue 002878

gold (dark) d28e00

gold (light) d2b000

Emory University Logo Guidelines
Transition-based AMR Parsing

<table>
<thead>
<tr>
<th>Action</th>
<th>Current state ⇒ Result state</th>
<th>Assign labels</th>
<th>Precondition</th>
</tr>
</thead>
<tbody>
<tr>
<td>NEXT EDGE-(l_r)</td>
<td>((\sigma_0</td>
<td>\sigma', \beta_0</td>
<td>\beta', G) \Rightarrow (\sigma_0</td>
</tr>
<tr>
<td>SWAP-(l_r)</td>
<td>((\sigma_0</td>
<td>\sigma', \beta_0</td>
<td>\beta', G) \Rightarrow (\sigma_0</td>
</tr>
<tr>
<td>REATTACH(_k)-(l_r)</td>
<td>((\sigma_0</td>
<td>\sigma', \beta_0</td>
<td>\beta', G) \Rightarrow (\sigma_0</td>
</tr>
<tr>
<td>REPLACE HEAD</td>
<td>((\sigma_0</td>
<td>\sigma', \beta_0</td>
<td>\beta', G) \Rightarrow (\beta_0</td>
</tr>
<tr>
<td>REENTRANCE(_k)-(l_r)</td>
<td>((\sigma_0</td>
<td>\sigma', \beta_0</td>
<td>\beta', G) \Rightarrow (\sigma_0</td>
</tr>
<tr>
<td>MERGE</td>
<td>((\sigma_0</td>
<td>\sigma', \beta_0</td>
<td>\beta', G) \Rightarrow (\sigma_0</td>
</tr>
<tr>
<td>NEXT NODE-(l_c)</td>
<td>((\sigma_0</td>
<td>\sigma_1</td>
<td>\sigma', [], G) \Rightarrow (\sigma_1</td>
</tr>
<tr>
<td>DELETE NODE</td>
<td>((\sigma_0</td>
<td>\sigma_1</td>
<td>\sigma', [], G) \Rightarrow (\sigma_1</td>
</tr>
</tbody>
</table>