AMR Parsing:
JAMR (Flanigan et al., 2014), Latent Alignment (Lyu & Titov, 2018)

Austin Blodgett

AMR Parsing Task

1. Alignment
2. Concept Identification
3. Relation Identification
4. Postprocessing – *wikification, focus identification, normalizing graph*

JAMR (Flanigan et al., 2014)

1. Alignment *Rule based*
2. Concept Identification *Semi-Markov Model*
3. Relation Identification Maximum Spanning Subgraph
4. Postprocessing *almost none*

JAMR: Rule-based Alignment

- 1. (Named Entity) Applies to name concepts and their opn children. Matches a span that exactly matches its opn children in numerical order.

“District of Columbia”

(n / name :op1 "District" :op2 "of" :op3 "Colombia")

JAMR: Rule-based Alignment

- 2. (Fuzzy Named Entity) Applies to name concepts and their :opn children. Matches a span that matches the fuzzy match of each child in numerical order.

"New Yorker"

(n / <u>name</u> :op1 "New" :op2 "York")

JAMR: Rule-based Alignment

- 3. (Date Entity) Applies to date-entity concepts and their day, month, year children (if exist). Matches any permutation of day, month, year, (two digit or four digit years), with or without spaces.

“11 15 2018”

(date-entity :month 11 :day 15 :year 2018)

JAMR: Rule-based Alignment

- 4. (Minus Polarity Tokens) Applies to - concepts, and matches “no”, “not”, “non.”

“no”

:polarity -

JAMR: Rule-based Alignment

- 5. (Single Concept) Applies to any concept. Strips off trailing '-[0-9]+' from the concept (for example run-01 → run), and matches any exact matching word or WordNet lemma.

"run"

run-01

JAMR: Rule-based Alignment

- 6. (Fuzzy Single Concept) Applies to any concept. Strips off trailing ‘-[0-9]+’, and matches the fuzzy match of the concept.

“wants”

want-01

JAMR: Rule-based Alignment

- 7. (U.S.) Applies to name if its op1 child is united and its op2 child is states. Matches a word that matches “us”, “u.s.” (no space), or “u. s.” (with space).

“U.S.”

(c/country :name (n/name :op1 “United” :op2 “States”))

JAMR: Rule-based Alignment

- 8. (Entity Type) Applies to concepts with an outgoing name edge whose head is an aligned fragment. Updates the fragment to include the unaligned concept. Ex: continent in (continent :name (name :op1 "Asia")) aligned to “asia.”

“Asia”

(continent :name (name :op1 "Asia"))

JAMR: Rule-based Alignment

- 9.(Quantity) Applies to .*-quantity concepts with an outgoing unit edge whose head is aligned. Updates the fragment to include the unaligned concept. Ex: distance-quantity in (distance-quantity :unit kilometer) aligned to "kilometres."

"kilometres"

(distance-quantity :unit kilometer)

JAMR: Rule-based Alignment

- 10. (Person-Of, Thing-Of) Applies to person and thing concepts with an outgoing .*-of edge whose head is aligned. Updates the fragment to include the unaligned concept. Ex: person in (person :ARG0-of strike-02) aligned to "strikers."

"writer"

(person :ARG0-of write-01)

JAMR: Rule-based Alignment

- 11. (Person) Applies to person concepts with a single outgoing edge whose head is aligned. Updates the fragment to include the unaligned concept. Ex: person in (person :poss (country :name (name :op1 "Korea")))

"Korean"

(person :poss (country :name (name :op1 "Korea")))

JAMR: Rule-based Alignment

- 12. (Goverment Organization) Applies to concepts with an incoming ARG.*-of edge whose tail is an aligned government-organization concept. Updates the fragment to include the unaligned concept. Ex: govern-01 in (government-organization :ARG0-of govern-01) aligned to “government.”

“government”

(government-organization :ARG0-of govern-01)

JAMR: Rule-based Alignment

- 13. (Minus Polarity Prefixes) Applies to - concepts with an incoming polarity edge whose tail is aligned to a word beginning with “un”, “in”, or “il.” Updates the fragment to include the unaligned concept. Ex: - in (employ-01 :polarity -) aligned to “unemployment.”

“illegal”

(l/legal :polarity -)

JAMR: Rule-based Alignment

- 14. (Degree) Applies to concepts with an incoming degree edge whose tail is aligned to a word ending is “est.” Updates the fragment to include the unaligned concept. Ex: most in (large :degree most) aligned to “largest.”

“largest”

(large :degree most)

JAMR: Concept Identification

- JAMR uses a semi-Markov Model:

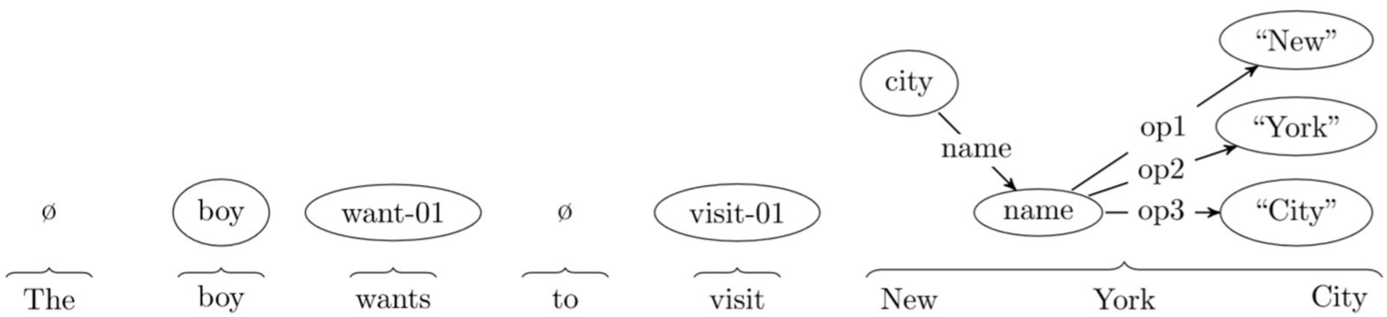


Figure 2: A concept labeling for the sentence "The boy wants to visit New York City."

JAMR: Concept Identification

Score based on these features:

- **Fragment given words**: Relative frequency estimates of the probability of a concept graph fragment given the sequence of words in the span. This is calculated from the concept-word alignments in the training corpus (§5).
- **Length of the matching span** (number of tokens).
- **NER**: 1 if the named entity tagger marked the span as an entity, 0 otherwise.
- **Bias**: 1 for any concept graph fragment from F and 0 for ∅.

JAMR: Relation Identification

- **JAMR uses a (novel) algorithm to find the Maximum Spanning Connected Subgraph**
  - **Start with all possible labelled edges**
  - **Score each edge using a discriminative model**
  - **Add edges in the right order to get desired properties of graph (connected, spanning, etc.)**

JAMR: Relation Identification

A solution should be:

1. **Preserving:** all graph fragments (including labels) from the concept identification phase are subgraphs of G.
2. **Simple:** only one edge between any two nodes
3. **Connected**
4. **Deterministic:** only one of each label per source node

JAMR: Relation Identification

input : weighted, connected graph $\langle V, E \rangle$ and set of edges $E^{(0)} \subseteq E$ to be preserved
output: maximum spanning, connected subgraph of $\langle V, E \rangle$ that preserves $E^{(0)}$

let $E^{(1)} = E^{(0)} \cup \{e \in E \mid \boldsymbol{\psi}^\top \mathbf{g}(e) > 0\}$;
create a priority queue Q containing $\{e \in E \mid \boldsymbol{\psi}^\top \mathbf{g}(e) \leq 0\}$ prioritized by scores;
$i = 1$;
while *Q nonempty and $\langle V, E^{(i)} \rangle$ is not yet spanning and connected* **do**
  $i = i + 1$;
  $E^{(i)} = E^{(i-1)}$;
  $e = \arg\max_{e' \in Q} \boldsymbol{\psi}^\top \mathbf{g}(e')$;
  remove e from Q;
  if *e connects two previously unconnected components of $\langle V, E^{(i)} \rangle$* **then**
    add e to $E^{(i)}$
  end
end
return $G = \langle V, E^{(i)} \rangle$;

Algorithm 1: MSCG algorithm.

JAMR: Relation Identification

Name	Description
Label	For each $\ell \in L_E$, 1 if the edge has that label
Self edge	1 if the edge is between two nodes in the same fragment
Tail fragment root	1 if the edge's tail is the root of its graph fragment
Head fragment root	1 if the edge's head is the root of its graph fragment
Path	Dependency edge labels and parts of speech on the shortest syntactic path between any two words in the two spans
Distance	Number of tokens (plus one) between the two concepts' spans (zero if the same)
Distance indicators	A feature for each distance value, that is 1 if the spans are of that distance
Log distance	Logarithm of the distance feature plus one.
Bias	1 for any edge.

Table 1: Features used in relation identification. In addition to the features above, the following conjunctions are used (Tail and Head concepts are elements of L_V): Tail concept $\wedge$ Label, Head concept $\wedge$ Label, Path $\wedge$ Label, Path $\wedge$ Head concept, Path $\wedge$ Tail concept, Path $\wedge$ Head concept $\wedge$ Label, Path $\wedge$ Tail concept $\wedge$ Label, Path $\wedge$ Head word, Path $\wedge$ Tail word, Path $\wedge$ Head word $\wedge$ Label, Path $\wedge$ Tail word $\wedge$ Label, Distance $\wedge$ Label, Distance $\wedge$ Path, and Distance $\wedge$ Path $\wedge$ Label. To conjoin the distance feature with anything else, we multiply by the distance.

JAMR: Focus Identification

- Every AMR has a Focus (top node in AMR parse)
- JAMR finds the focus as a part of concept identification by the following
  - Add a root node and “focus” edges
  - Require that there be one edge from root
  - Identify the target node as the focus of the AMR

JAMR: Results

Models	A' 17	C' 16	J' 16	Ch' 17	Ours
Dataset	R1	R1	R1	R2	R2
Smatch	64	63	67	71	**74.4**±0.16
Unlabeled	69	69	69	74	**77.1**±0.10
No WSD	65	64	68	72	**75.5**±0.12
Reentrancy	41	41	42	**52**	**52.3**±0.43
Concepts	83	80	83	82	**85.9**±0.11
NER	83	75	79	79	**86.0**±0.46
Wiki	64	0	75	65	**75.7**±0.30
Negations	48	18	45	**62**	58.4±1.32
SRL	56	60	60	66	**69.8**±0.24

Table 2: F1 scores on individual phenomena. A'17 is AMREager, C'16 is CAMR, J'16 is JAMR, Ch'17 is ChSeq+100K. Ours are marked with standard deviation.

Latent Alignment (Lyu & Titov, 2018)

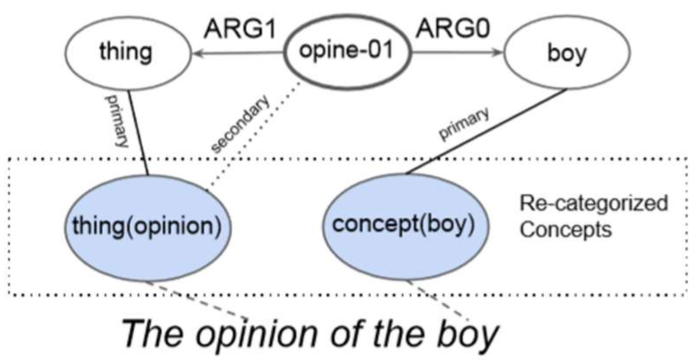
1. Alignment	*Probabilistic, Latent*
2. Concept Identification	*RNN*
3. Relation Identification	*concepts+word embeddings FFNN*
4. Postprocessing	*wikification, word sense disambiguation, etc.*

Latent Alignment: Concept Identification

- RNN (BiLSTM) model predicts either a concept or *Null*
  - Word can only map to one concept
  - Deal with larger mappings by recategorizing AMR

Latent Alignment: Concept Identification

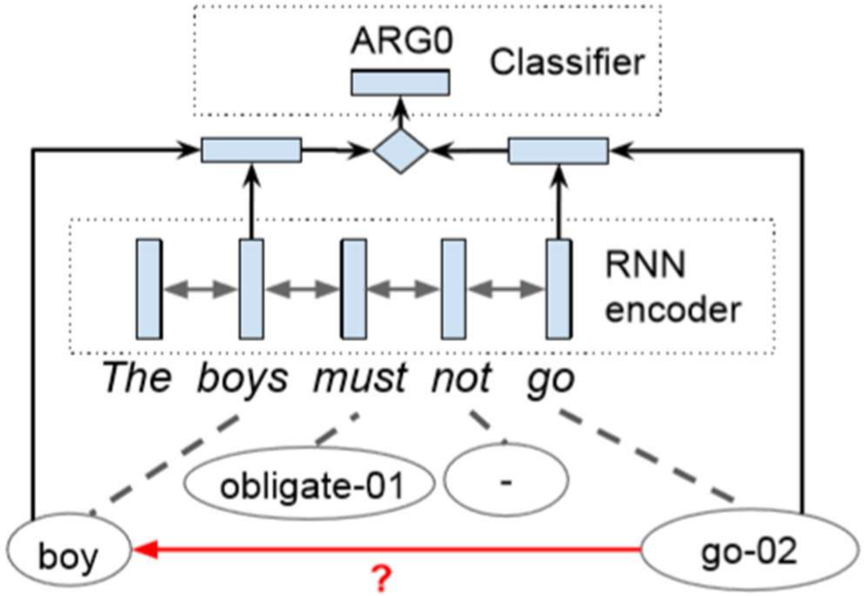
- Recategorize AMR subgraphs



Latent Alignment: Relation Identification

- To predict relation
  - Use source and target concept embeddings
  - Use RNN states for source and target words
  - Use log-linear classifier
  - Predict either relation name or *Null*

Latent Alignment: Relation Identification



Latent Alignment: Alignment

- **Model Latent Alignment During Training**
- An Alignment is a permutation between
  - n Words
  - m Concepts + (n-m) Nulls
- Alignment can rely on sequence of concepts and RNN states
- Use RNN (BiLSTM) to predict probability of alignment
  - Create objective function using Gumbel-Sinkhorn, which estimates samples from the global distribution

Latent Alignment: Results

Models	A' 17	C' 16	J' 16	Ch' 17	Ours
Dataset	R1	R1	R1	R2	R2
Smatch	64	63	67	71	**74.4**±0.16
Unlabeled	69	69	69	74	**77.1**±0.10
No WSD	65	64	68	72	**75.5**±0.12
Reentrancy	41	41	42	**52**	**52.3**±0.43
Concepts	83	80	83	82	**85.9**±0.11
NER	83	75	79	79	**86.0**±0.46
Wiki	64	0	75	65	**75.7**±0.30
Negations	48	18	45	**62**	58.4±1.32
SRL	56	60	60	66	**69.8**±0.24

Table 2: F1 scores on individual phenomena. A'17 is AMREager, C'16 is CAMR, J'16 is JAMR, Ch'17 is ChSeq+100K. Ours are marked with standard deviation.