Unsupervised AMR-Dependency Parse Alignment

Wei-Te Chen Martha Palmer

Department of Computer Science
University of Colorado Boulder

weite.chen@colorado.edu

April 6th, 2017

Computational Language and EducAtion Research
CLEAR

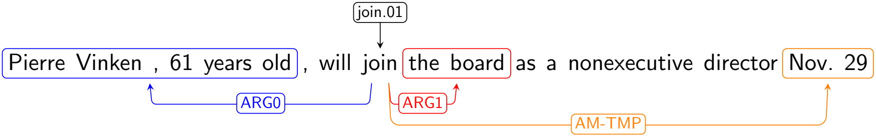
University of Colorado Boulder

How to Represent "Meaning"

- First-Order Logical Form

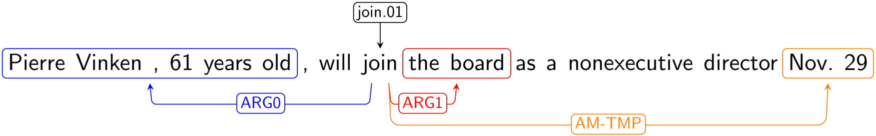
How to Represent "Meaning"

- First-Order Logical Form
- Semantic Role Labeling (SRL)



How to Represent “Meaning”

- First-Order Logical Form
- Semantic Role Labeling (SRL)



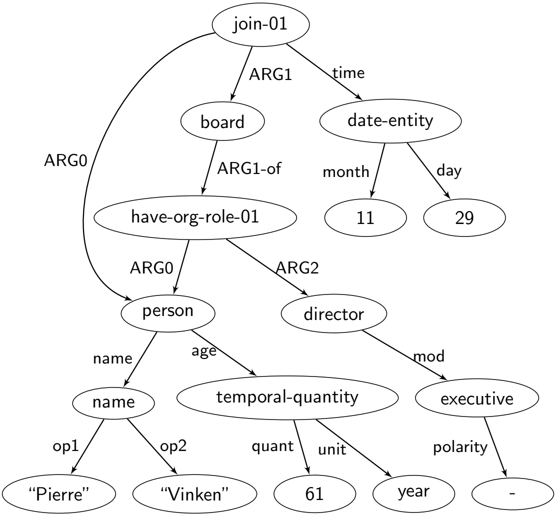
- **Abstract Meaning Representation (AMR)**

```
(j / join-01
   :ARG0 (p / person :wiki -
     :name (p2 / name :op1 "Pierre" :op2 "Vinken")
     :age (t / temporal-quantity :quant 61 :unit (y / year)))
   :ARG1 (b / board
     :ARG1-of (h / have-org-role-91
       :ARG0 p :ARG2 (d2 / director
         :mod (e / executive :polarity - ))))
   :time (d / date-entity :month 11 :day 29))
```

Abstract Meaning Representation (AMR)

AMR is a semantic representation that expresses the meaning of a sentence

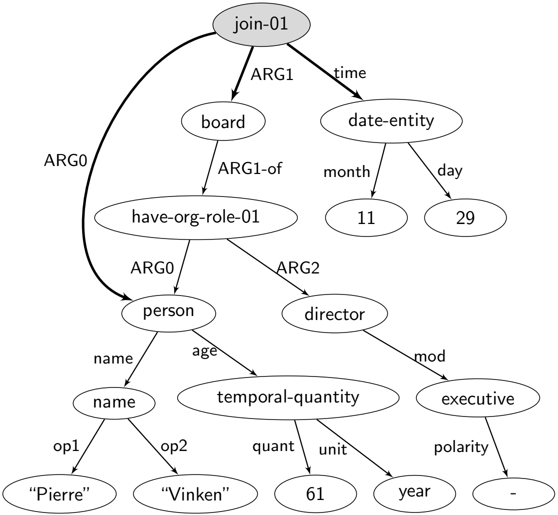
- A rooted, acyclic graph



Abstract Meaning Representation (AMR)

AMR is a semantic representation that expresses the meaning of a sentence

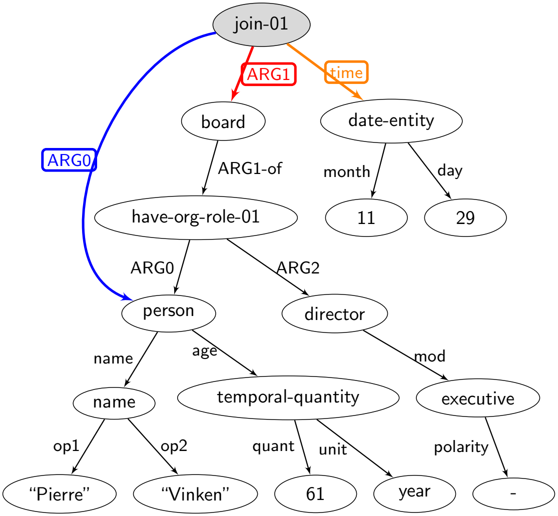
- A rooted, acyclic graph



Abstract Meaning Representation (AMR)

AMR is a semantic representation that expresses the meaning of a sentence

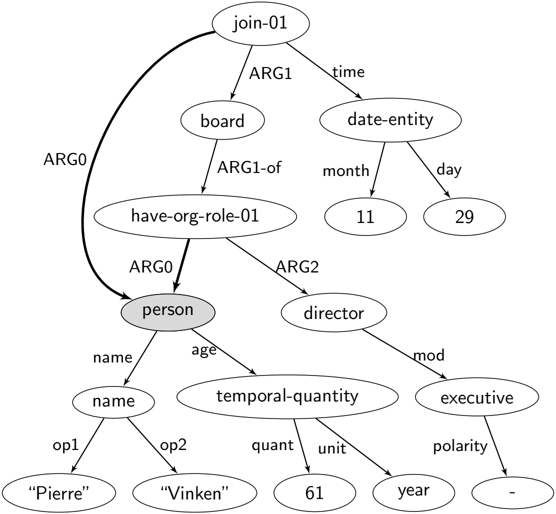
- Relies heavily on predicate-argument structures from PropBank



Abstract Meaning Representation (AMR)

AMR is a semantic representation that expresses the meaning of a sentence

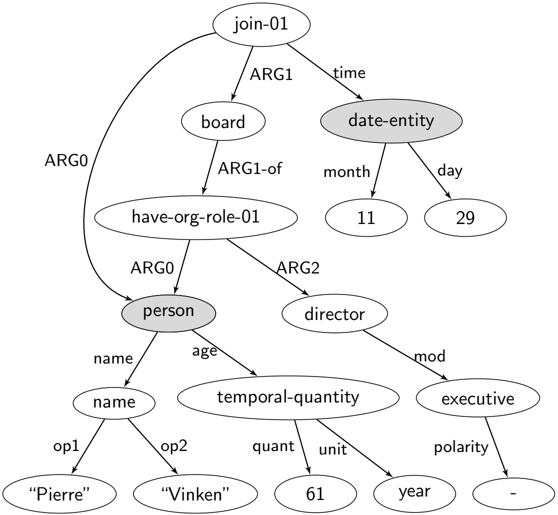
- Uses reentrance to represent co-reference



Abstract Meaning Representation (AMR)

AMR is a semantic representation that expresses the meaning of a sentence

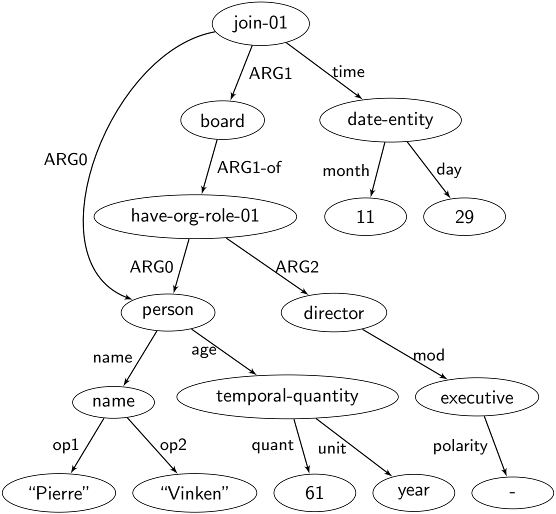
- Encodes named entities, wiki-links, and discourse connectives



Abstract Meaning Representation (AMR)

AMR is a semantic representation that expresses the meaning of a sentence

- Abstracts away from syntactic idiosyncrasies



Review: AMR Parsing

Graph-based AMR Parser

Transition-Based AMR Parser

Review: AMR Parsing

Graph-based AMR Parser

- Separate parsing task into concept identification and relation identification

Transition-Based AMR Parser

Review: AMR Parsing

Graph-based AMR Parser

- Separate parsing task into concept identification and relation identification
- Aim to find a connected graph with a maximum sum of edge (relation) scores

Transition-Based AMR Parser

Review: AMR Parsing

Graph-based AMR Parser

- Separate parsing task into concept identification and relation identification
- Aim to find a connected graph with a maximum sum of edge (relation) scores

Transition-Based AMR Parser

- Generate AMR graphs through conversion from dependency parse trees

Review: AMR Parsing

Graph-based AMR Parser

- Separate parsing task into concept identification and relation identification
- Aim to find a connected graph with a maximum sum of edge (relation) scores

Transition-Based AMR Parser

- Generate AMR graphs through conversion from dependency parse trees
- Design different parsing actions

Review: AMR Parsing

Graph-based AMR Parser

- Separate parsing task into concept identification and relation identification
- Aim to find a connected graph with a maximum sum of edge (relation) scores

Transition-Based AMR Parser

- Generate AMR graphs through conversion from dependency parse trees
- Design different parsing actions
- State-of-the-art system: CAMR (Wang et al. 2015a, 2015b) - F_1 : **0.62**

Review: AMR Parsing

Graph-based AMR Parser

- Separate parsing task into concept identification and relation identification
- Aim to find a connected graph with a maximum sum of edge (relation) scores

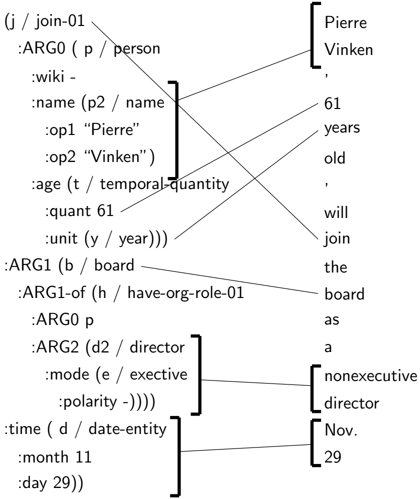
Transition-Based AMR Parser

- Generate AMR graphs through conversion from dependency parse trees
- Design different parsing actions
- State-of-the-art system: CAMR (Wang et al. 2015a, 2015b) - F_1 : **0.62**

No gold standard word-concept mappings

Review: AMR Aligner

Aligner Strategy I: **Align AMR concepts and relations to spans of words**



Review: AMR Aligner

Aligner Strategy I: **Align AMR concepts and relations to spans of words**

Heuristic Aligner

- JAMR (Flanigan et al., 2014)

```
(j / join-01
  :ARG0 ( p / person
    :wiki -
    :name (p2 / name
      :op1 “Pierre”
      :op2 “Vinken”)
    :age (t / temporal-quantity
      :quant 61
      :unit (y / year)))
:ARG1 (b / board
  :ARG1-of (h / have-org-role-01
    :ARG0 p
    :ARG2 (d2 / director
      :mode (e / exective
        :polarity -))))
:time ( d / date-entity
  :month 11
  :day 29))
```

Pierre
Vinken
,
61
years
old
,
will
join
the
board
as
a
nonexecutive
director
Nov.
29

Review: AMR Aligner

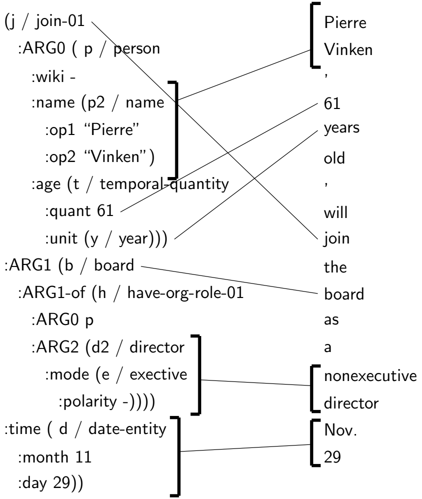
Aligner Strategy I: **Align AMR concepts and relations to spans of words**

Heuristic Aligner

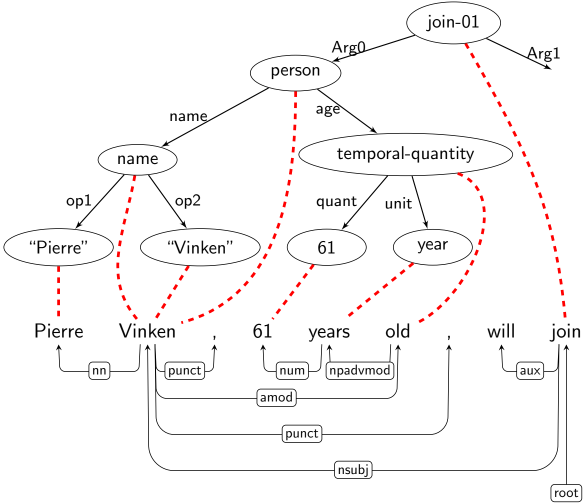
- JAMR (Flanigan et al., 2014)

Unsupervised Aligner

- ISI Aligner (Pourdamghani et al., 2014)
- Stanford Aligner (Werling et al., 2015)

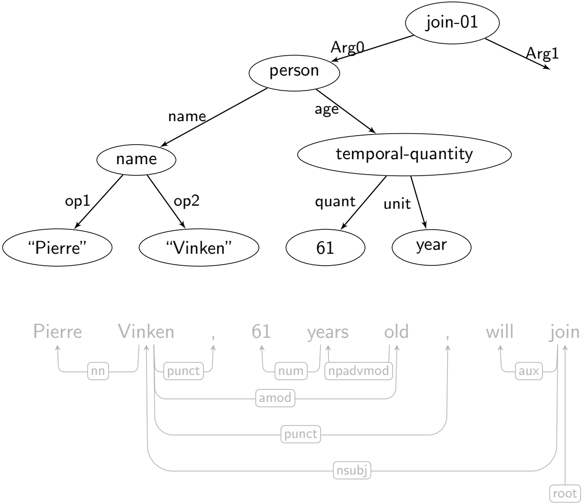


Aligner Strategy II: **Aligns AMR concepts and relations to word nodes in a dependency parse tree**



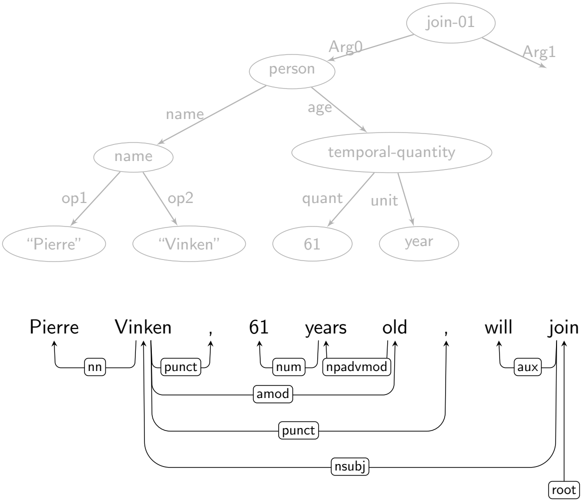
AMR - Dependency Parse Aligner

Aligner Strategy II: **Aligns AMR concepts and relations to word nodes in a dependency parse tree**



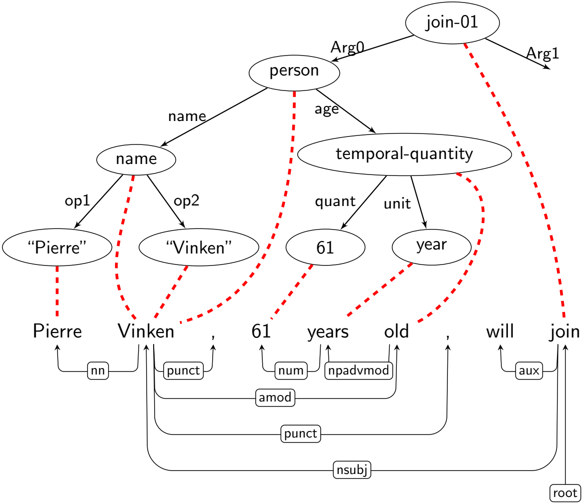
AMR - Dependency Parse Aligner

Aligner Strategy II: **Aligns AMR concepts and relations to word nodes in a dependency parse tree**



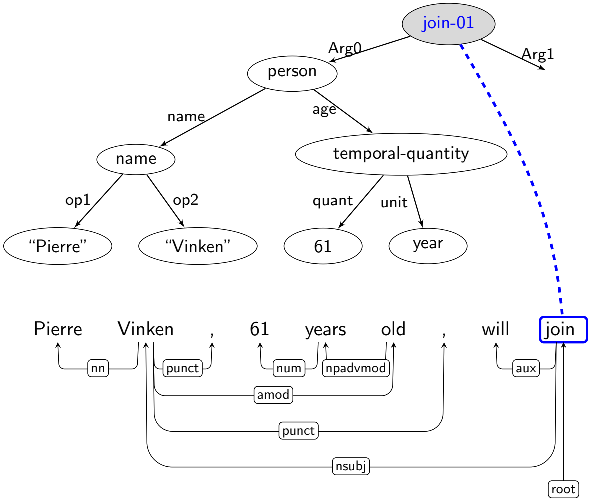
AMR - Dependency Parse Aligner

Aligner Strategy II: **Aligns AMR concepts and relations to word nodes in a dependency parse tree**



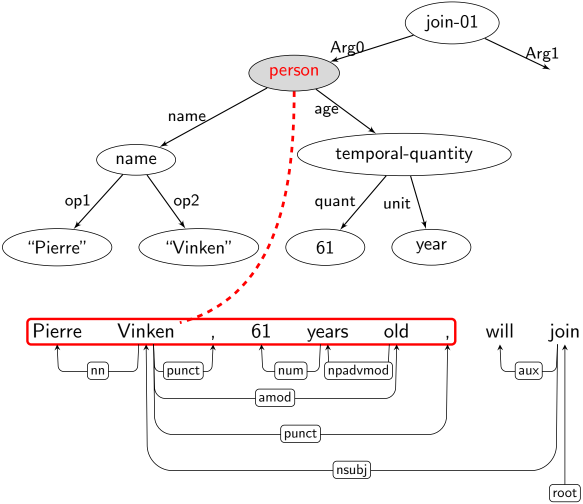
AMR - Dependency Parse Aligner

Aligner Strategy II: **Aligns AMR concepts and relations to word nodes in a dependency parse tree**



AMR - Dependency Parse Aligner

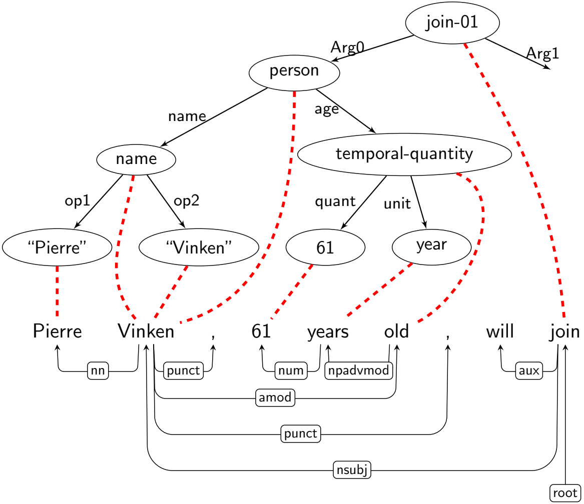
Aligner Strategy II: **Aligns AMR concepts and relations to word nodes in a dependency parse tree**



AMR - Dependency Parse Aligner

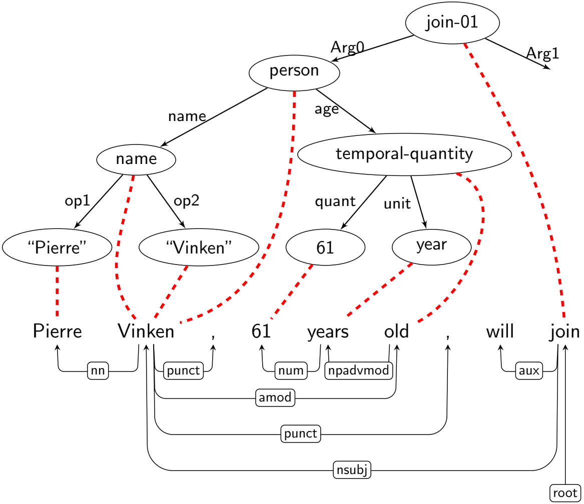
Aligner Strategy II: **Aligns AMR concepts and relations to word nodes in a dependency parse tree**

- AMR concept → dependency parse node (one-to-one) alignment



Aligner Strategy II: **Aligns AMR concepts and relations to word nodes in a dependency parse tree**

- AMR concept → dependency parse node (one-to-one) alignment
- Aim to find better alignments to benefit AMR parsing



Training

Objective Function

$$\Theta = \operatorname{argmax} L_{\Theta}(\text{AMR}|\text{DEP}) \tag{1}$$

$$L_{\Theta}(\text{AMR}|\text{DEP}) = \prod_{(C,D,A)\in\mathbb{S}} P(C|D) = \prod_{(C,D,A)\in\mathbb{S}} \sum_{a\in A} P(C,a|D) \tag{2}$$

where

- $\mathbb{S}$: training samples
- C: AMR
- D: dependency parse
- A: all alignment set between C and D
- a: alignment function

Training with EM Algorithm: E-Step

E-Step estimates all alignment probabilities of a (C, D) pair (in Eq. (3))

- By giving the product of feature probabilities (in Eq. (4))

$$P(a|C,D) = \prod_{j=1}^{|C|} \frac{P(c_j|d_{c_j} = a(c_j), d_{c_j^p} = a(c_j^p))}{\sum_{l=1}^{|D|} \sum_{i=1}^{|D|} P(c_j|d_i, d_l)} \tag{3}$$

$$P(c_j|d_i, d_l) = \prod_{\theta \in \Theta} P_\theta(c_j, d_i, d_l) \tag{4}$$

where

- c_j: j-th concept in C
- c_j^p: parent node of c_j
- d_c: dependency node aligned by c

Training with EM Algorithm: M-Step

In M-Step, feature probabilities are re-estimated by collecting the count of all AMR-dependency parse pairs

Collect Count

$$cnt_\theta(c|d_c, d_{c^p}; C, D) = \sum_{a \in A} \frac{P(c|d_c, d_{c_p})}{\sum_{i=0}^{|D|} \sum_{l=0}^{|D|} P(c|d_i, d_l)} \tag{5}$$

Update Probability

$$P_\theta(c, d, d^p) \leftarrow \sum_{C \in AMR, D \in DEP} \frac{cnt_\theta(c|d_c, d_{c^p}; C, D)}{\sum_c cnt_\theta(c|d_c, d_{c^p}; C, D)} \tag{6}$$

Features

- Basic Features
- External Features
  - Lemma
  - Relation
  - Named Entity
  - Semantic Role
- Global Feature

Feature: Basic Match Type

Basic Match Type

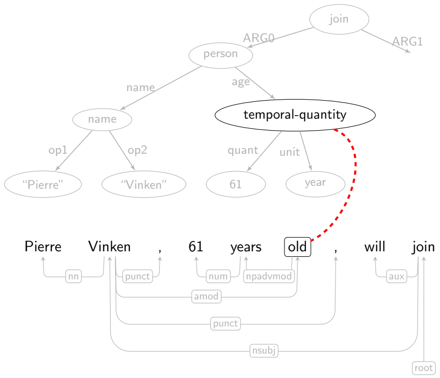
- **Word Form** e.g. "join-01" aligns to *join*
- **Numbers**, **Ordinal Numbers**, **Date**

	Match Type	at Concept	at Leaf
(1)	Word	45.2%	73.4%
(2)	Word (case insensitive)	-	0.9%
(3)	Lemma (case insensitive)	10.8%	0.3%
(4)	Partial match with word	6.1%	8.2%
(5)	Partial match with lemma	0.2%	0.3%
(6)	Numbers	-	3.1%
(7)	Ordinal Numbers	-	2.8%
(8)	Date	-	4.3%
(9)	Others	37.7%	6.5%

Table 1: The rules and distribution of basic match types

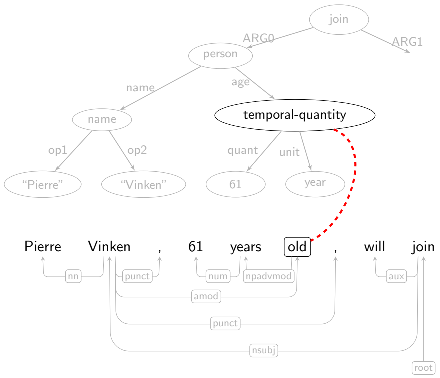
Lemma Probability

- $P_{Lemma}(c, d_c) = P(c|Word(d_c))$



Lemma Probability

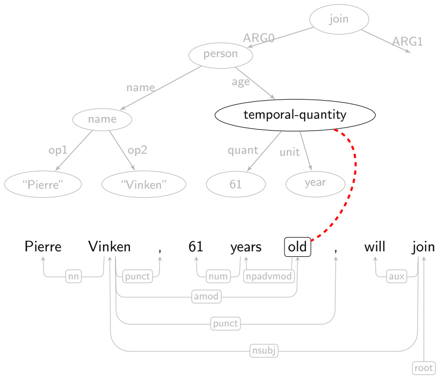
- $P_{Lemma}(c, d_c) = P(c|Word(d_c))$



$P_{Lemma}(c = \text{temporal-quanity}, d_c = \underline{old})$

Lemma Probability

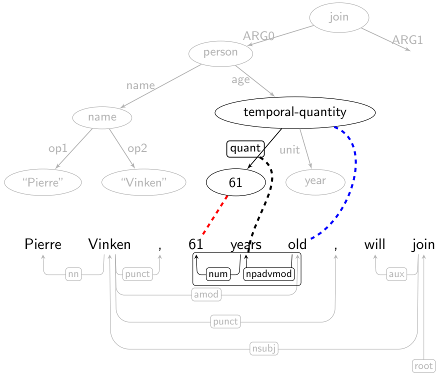
- $P_{Lemma}(c, d_c) = P(c|Word(d_c))$



$P_{Lemma}(c = \text{temporal-quanity}, d_c = \underline{old})$
$= P(\text{temporal-quanity}|Word(\underline{old}))$

Relation Probability

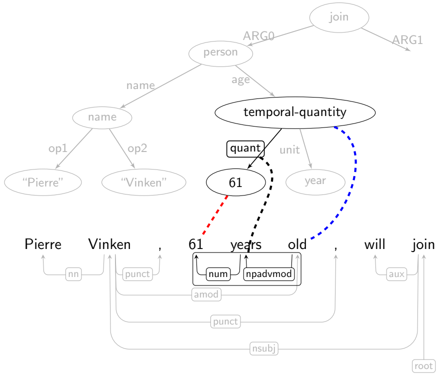
- $P_{rel}(c, d_c, d_{c^p}) = P(AMRLabel(c)|Path(d_c, d_{c^p}))$



External Features - Relation Probability

Relation Probability

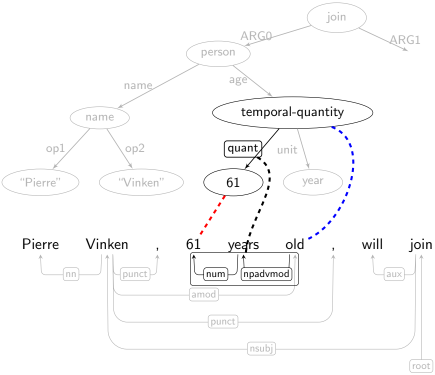
- $P_{rel}(c, d_c, d_{c^p}) = P(AMRLabel(c) | Path(d_c, d_{c^p}))$



$P_{rel}(c = 61, d_c = \underline{61}, d_{c^p} = \underline{old})$

Relation Probability

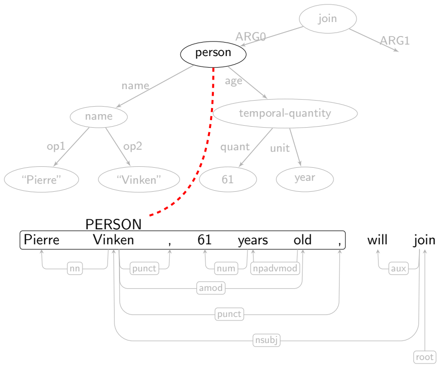
- $P_{rel}(c, d_c, d_{c^p}) = P(AMRLabel(c)|Path(d_c, d_{c^p}))$



$P_{rel}(c = 61, d_c = \underline{61}, d_{c^p} = \underline{old})$
$= P(quant|\underline{advmod \downarrow num \downarrow})$

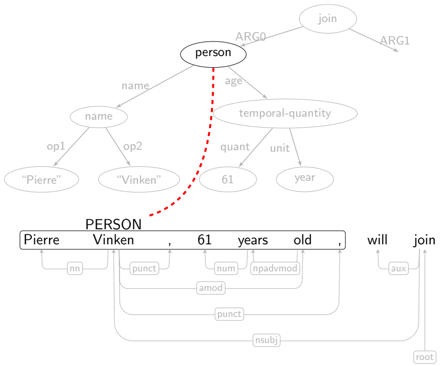
Named Entity Probability

- $P_{NE}(c, d_c) = P(c|NamedEntity(d_c))$



Named Entity Probability

- $P_{NE}(c, d_c) = P(c|NamedEntity(d_c))$

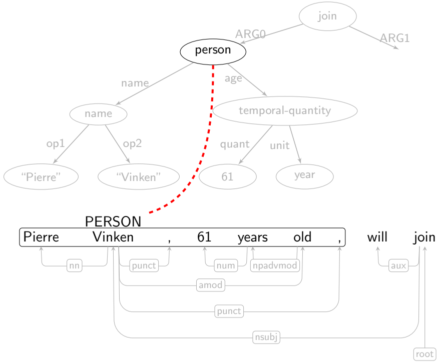


$P_{NE}(c = \text{person}, d_c = \underline{Vinken})$

External Features - Named Entity Probability

Named Entity Probability

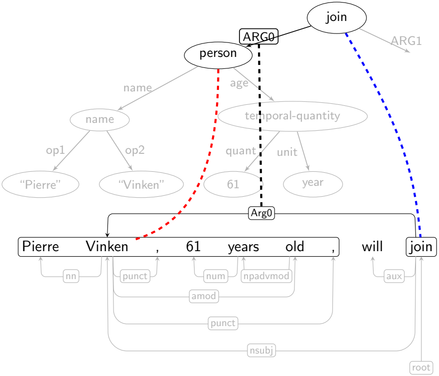
- $P_{NE}(c, d_c) = P(c|NamedEntity(d_c))$



$P_{NE}(c = \text{person}, d_c = \underline{Vinken})$
$= P(\text{person}|\underline{PERSON})$

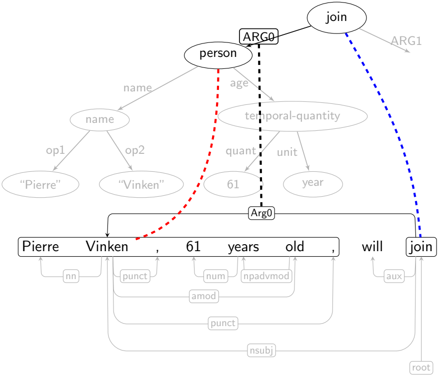
Semantic Role Probability

- $P_{SR}(c, d_c, d_{c^p}) = P(AMRLabel(c) | Role(d_{c^p}, d_c))$



Semantic Role Probability

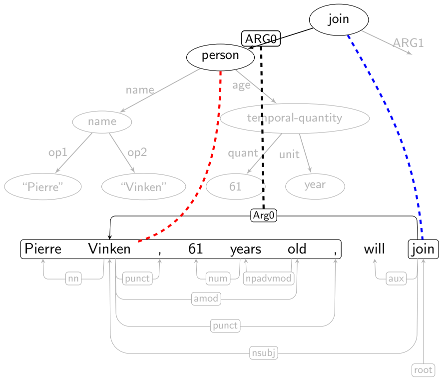
- $P_{SR}(c, d_c, d_{c^p}) = P(AMRLabel(c) | Role(d_{c^p}, d_c))$



$P_{SR}(c = \text{person}, d_c = \underline{Vinken}, d_{c^p} = \underline{join})$

Semantic Role Probability

- $P_{SR}(c, d_c, d_{c^p}) = P(AMRLabel(c)|Role(d_{c^p}, d_c))$



$$P_{SR}(c = \text{person}, d_c = \underline{Vinken}, d_{c^p} = \underline{join})$$
$$= P(\text{ARG0}|\underline{Arg0})$$

Global Feature

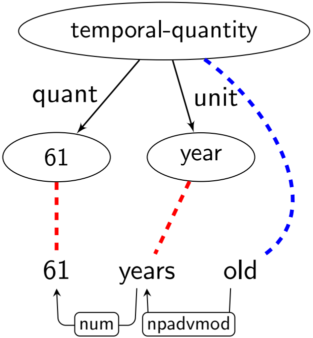
To ensure that parent concept is aligned to phrase which contains the sub-phrase that aligned by child concept, $R_{CC}(c^p)$ is designed.

- Good Alignment
- Bad Alignment

Global Feature

To ensure that parent concept is aligned to phrase which contains the sub-phrase that aligned by child concept, $R_{CC}(c^p)$ is designed.

- Good Alignment

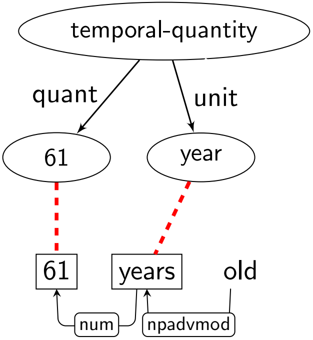


- Bad Alignment

Global Feature

To ensure that parent concept is aligned to phrase which contains the sub-phrase that aligned by child concept, $R_{CC}(c^p)$ is designed.

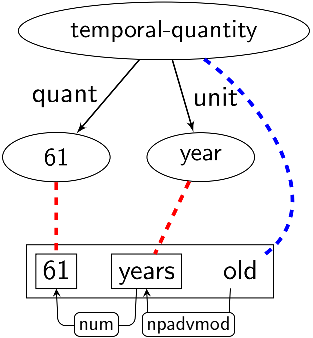
- Good Alignment



- Bad Alignment

To ensure that parent concept is aligned to phrase which contains the sub-phrase that aligned by child concept, $R_{CC}(c^p)$ is designed.

- Good Alignment

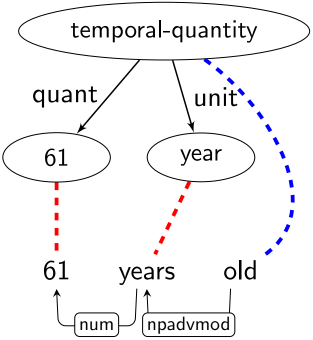


- Bad Alignment

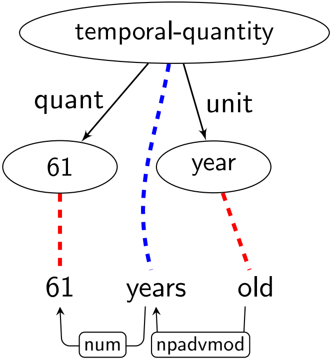
Global Feature

To ensure that parent concept is aligned to phrase which contains the sub-phrase that aligned by child concept, $R_{CC}(c^p)$ is designed.

- Good Alignment



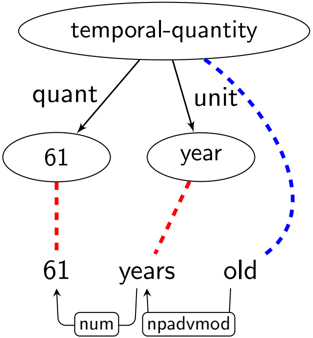
- Bad Alignment



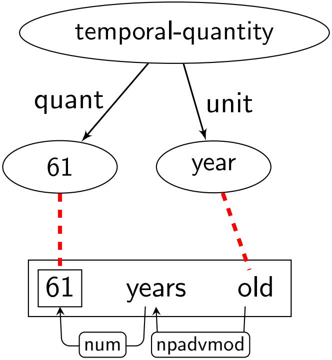
Global Feature

To ensure that parent concept is aligned to phrase which contains the sub-phrase that aligned by child concept, $R_{CC}(c^p)$ is designed.

- Good Alignment

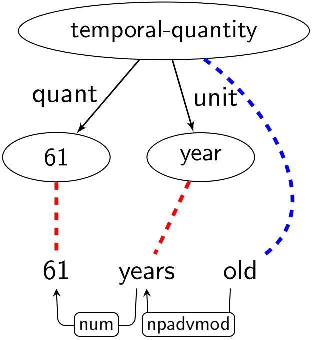


- Bad Alignment

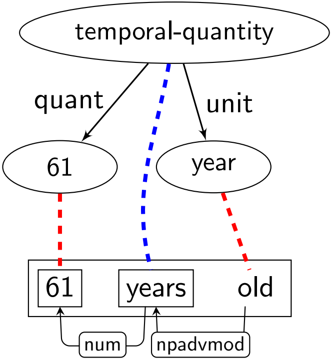


To ensure that parent concept is aligned to phrase which contains the sub-phrase that aligned by child concept, $R_{CC}(c^p)$ is designed.

- Good Alignment



- Bad Alignment

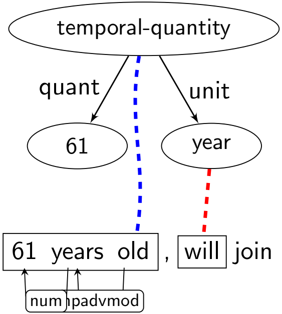


Global Feature

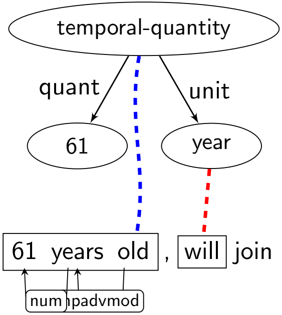
- Penalty

Global Feature

- Penalty



- Penalty



- Ration Score: R_{CC}

$$R_{CC}(c) = \frac{|W_{child}(c) \cap W(c)|}{|W(c)|} \times penalty(c)$$

$$W(c) = d_c; W_{child}(c) = \bigcup_{c^{s_i} \in child(c)} d_{c^{s_i}}$$

$$penalty(c) = \\ exp(-|W_{child}(c) \setminus (W_{child}(c) \cap W(c))|)$$

Decoding

Search the most possible alignments

- Use beam search algorithm
- Start from leaf concepts, and walk through all concepts

$$\underset{a}{\operatorname{argmax}} P(a|C,D) = \underset{a}{\operatorname{argmax}} \prod_{j=1}^{|C|} R_{CC}(c_j) * P(c_j | d_{c_j} = a(c_j), d_{c_j^p} = a(c_j^p)) \quad (7)$$

- **Running Time**: $O(|b| * |C| * |D|^2)$
  where $|b|$ is the beam size

Data Preparation

Corpus

- **AMR Data**: The LDC DEFT Phase 2 AMR Annotation Release 1.0
- **Gold Standard Dependency Parse**: OntoNotes (ON) 5.0

Training Set

- **Gold Dep.**: Sentences appear in both AMR Release and ON 5.0
- **Auto Dep.**: All sentences in AMR Data with dependency parses generated by ClearNLP.

Test and Development Set

- Manually align the AMR concepts and dependency word nodes

		Sent.	**Token**
Train	Gold Dep.	8,276	176,422
	Auto Dep.	39,260	649,219
Dev.		409	8,695
Test		415	8,786

Aligner Results

Feature Contribution

Data	Feature	P	R	F_1
Gold Dep.	L	84.0	85.0	84.5
	L + S	**85.2**	**86.3**	**85.7**
	L + S + R	82.8	83.8	83.3
	L + S + R + N	80.9	81.9	81.4
Auto Dep.	L	84.9	85.4	85.1
	L + S	85.7	87.4	86.5
	L + S + R	85.8	87.7	86.7
	L + S + R + N	**86.3**	**88.0**	**87.1**

Incremental Feature Contributions for different features:

- *L*: lemma
- *R*: relation
- *N*: NE
- *S*: semantic role

Aligner results with different aligners

Data	Aligner	P	R	F-Score
Gold Dep.	Chen 2015	61.1	53.4	57.0
	JAMR	78.5	62.8	69.8
	ISI	78.6	71.4	74.9
	Ours	**85.2**	**86.3**	**85.7**
Auto Dep.	Chen 2015	62.4	55.5	58.7
	JAMR	80.2	65.9	72.4
	ISI	80.4	74.9	77.6
	Ours	**86.3**	**88.0**	**87.1**

* Our aligner achieves the best F1 score in both data sets since it is designed to align AMRs to dependency parses.

Aligner Results

Using alignments with CAMR Parser

Data	Aligner	P	R	F-Score	Diff
Gold Dep.	JAMR	62.2	61.0	61.1	+5.3
	ISI	65.3	63.9	64.5	+1.9
	Ours	**68.6**	**64.2**	**66.4**	
Auto Dep.	JAMR	64.2	63.0	63.1	+3.6
	ISI	66.1	65.1	65.6	+1.1
	Ours	**68.1**	**64.7**	**66.7**	

Error Analysis

Error Category

- **Automatic Parsing Errors** - 3.8%

Error Analysis

Error Category

- **Automatic Parsing Errors** - 3.8%
- **Long Distance Dependencies** - 14.2%

Error Analysis

Error Category

- **Automatic Parsing Errors** - 3.8%
- **Long Distance Dependencies** - 14.2%
- **Duplicate Words** - 17.4%

Error Analysis

Error Category

- **Automatic Parsing Errors** - 3.8%
- **Long Distance Dependencies** - 14.2%
- **Duplicate Words** - 17.4%
- **Meaning Coverage Errors** - 40.4%

Conclusion and Future Work

- Present an AMR-Dependency Parse aligner, which estimates the feature probabilities by running the EM algorithm
- Our aligner can be used directly by dependency parse to AMR style parser
- Latent probabilities (i.e. external feature probatilities) can also benefit the AMR Parser

Thank You!
Questions

AMR does help some practical tasks

- Event Extraction
  - Liberal Event Extraction and Event Schema Induction (Huang et al., 2016)
  - Incorporate AMR as semantic representation system to detect and represent event structures

Method	ERE: Trigger F_1 (%)			ERE: Arg F_1(%)			ACE: Trigger F_1 (%)			ACE: Arg F_1 (%)		
	P	R	F_1	P	R	F_1	P	R	F_1	P	R	F_1
LSTM	41.5	46.8	44.1	9.9	11.6	10.7	66.0	60	62.8	29.3	32.6	30.8
Joint	42.3	41.7	42.0	61.8	23.2	33.7	73.7	62.3	67.5	64.7	44.4	52.7
DMCNN	-	-	-	-	-	-	75.6	63.6	**69.1**	68.8	46.9	**53.5**
Liberal$_{PerfectAMR}$	79.8	50.5	**61.8**	48.9	32.9	**39.3**	-	-	-	-	-	-
Liberal$_{SystemAMR}$	88.5	42.6	57.5	47.6	30.0	36.8	80.7	50.1	61.8	51.9	39.4	44.8