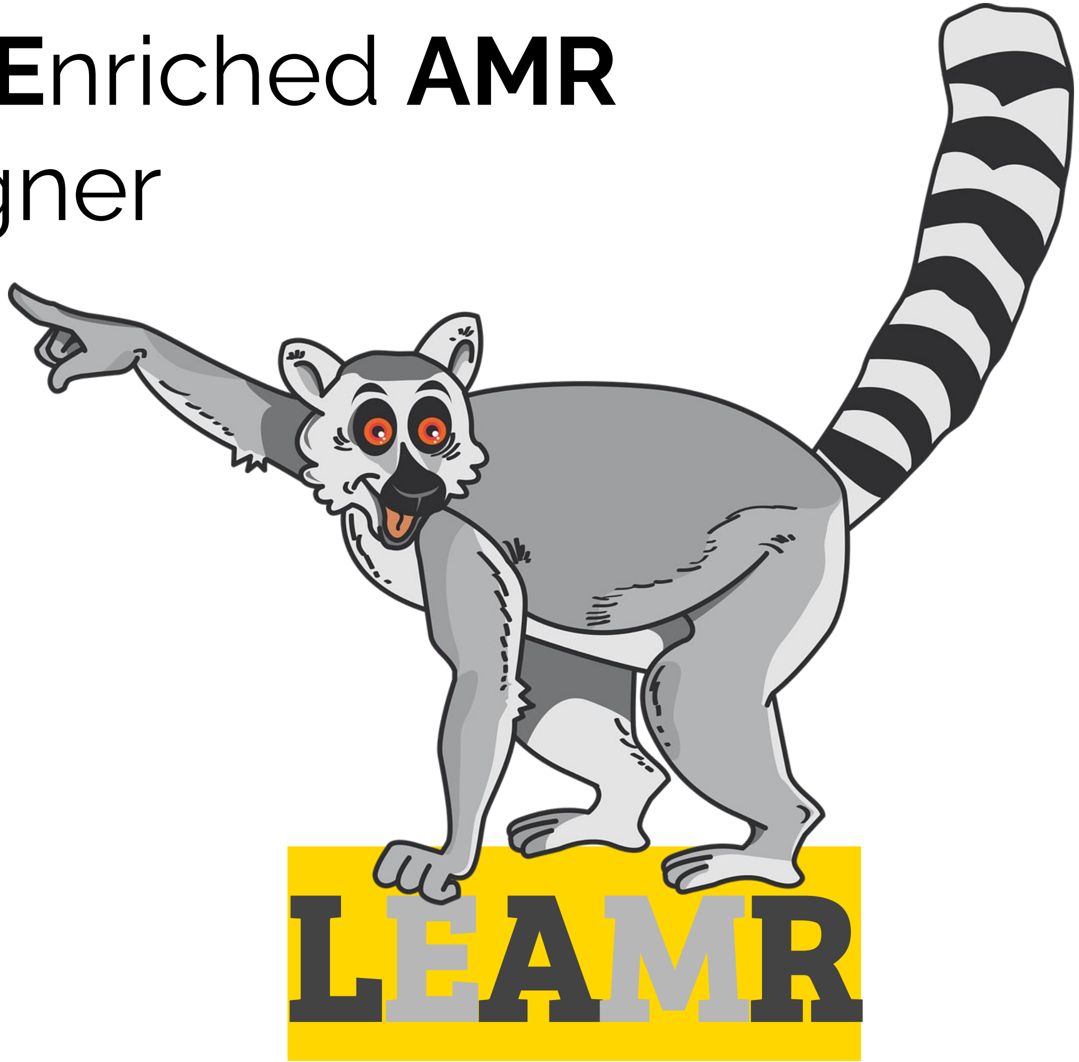
Probabilistic, Structure-Aware Algorithms for Improved Variety, Accuracy, and Coverage of AMR Alignments

Austin Blodgett & Nathan Schneider
Georgetown University
@ ACL 2021

Linguistically **E**nriched **AMR** Aligner



Contributions

- A novel *structurally-comprehensive* formulation of AMR-to-English alignment in terms of mappings between spans and connected subgraphs.
- **Released Data**: automatic data for AMR Release 3.0 and *Little Prince* data + 350 gold aligned sentences.
- **Alignment algorithm** which combines EM with rules. Advantages include:

   1. much improved ***coverage*** over previous datasets,
   2. increased ***variety*** of the substructures aligned, including alignments for all relations, and alignments for diagnosing reentrancies,
   3. alignments are made between ***spans*** and ***connected substructures*** of an AMR,
   4. broader ***identification of spans*** including named entities and verbal and prepositional multiword expressions.

Outline

1. AMR Alignment: Background
2. LEAMR: Multi-layer Formulation
3. Released Data
4. Algorithm
5. Results

Introduction: AMR

- Abstract Meaning Representation

captures “who did what to whom”

Introduction: AMR

- Abstract Meaning Representation

captures “who did what to whom”

Directed acyclic graph representation of sentence meaning

Introduction: AMR

- Abstract Meaning Representation

captures “who did what to whom”

Directed acyclic graph representation of sentence meaning

Consolidates a number of semantic prediction task:

- word sense disambiguation
- semantic role labelling
- named entity recognition
- coreference

Introduction: AMR

- Abstract Meaning Representation

captures “who did what to whom”

Directed acyclic graph representation of sentence meaning

Consolidates a number of semantic prediction task:
- word sense disambiguation
- semantic role labelling
- named entity recognition
- coreference

Scalable (~60,000 available English sentences)

Introduction: AMR

- Abstract Meaning Representation

captures “who did what to whom”

Directed acyclic graph representation of sentence meaning

Consolidates a number of semantic prediction task:

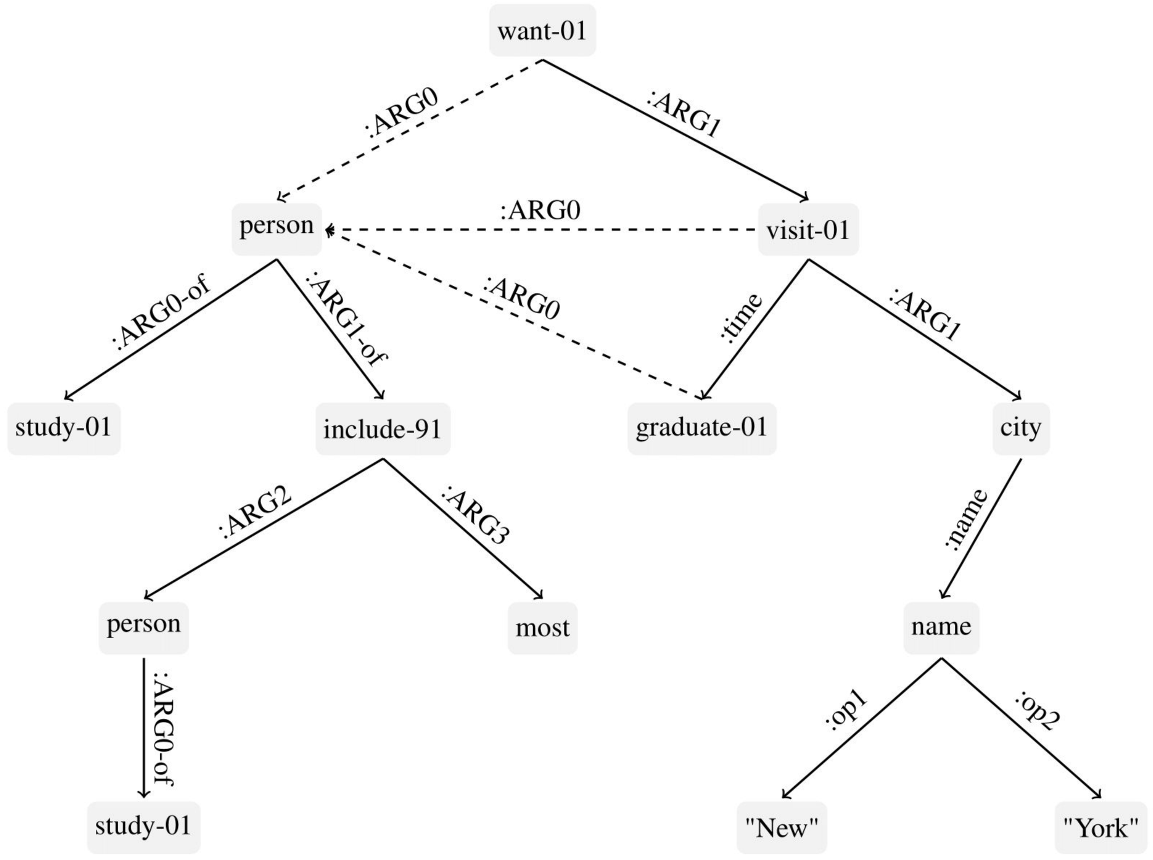
- word sense disambiguation
- semantic role labelling
- named entity recognition
- coreference

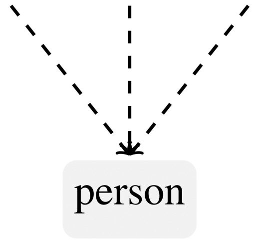
Scalable (~60,000 available English sentences)

Unanchored (lacks gold alignments)

Introduction: AMR

Reentrancies



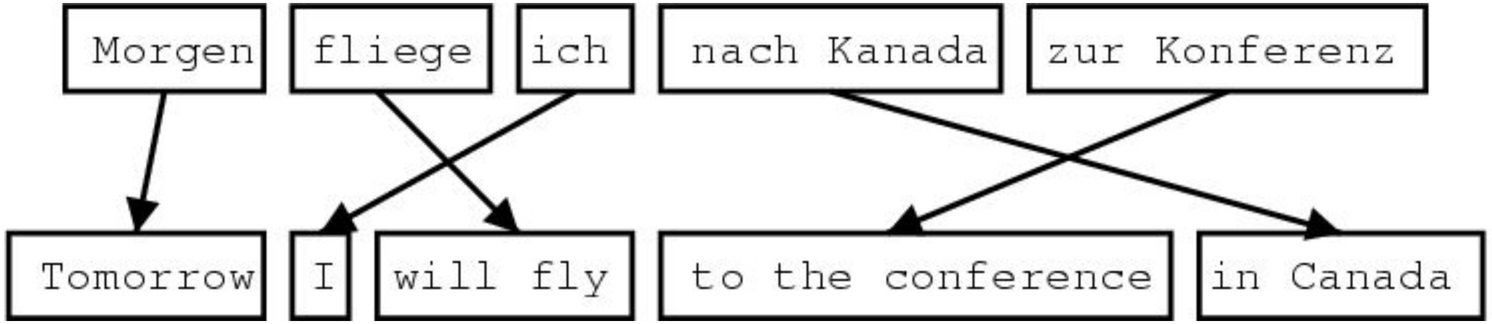


“Most of the students want to visit New York when they graduate”

Background: Alignment

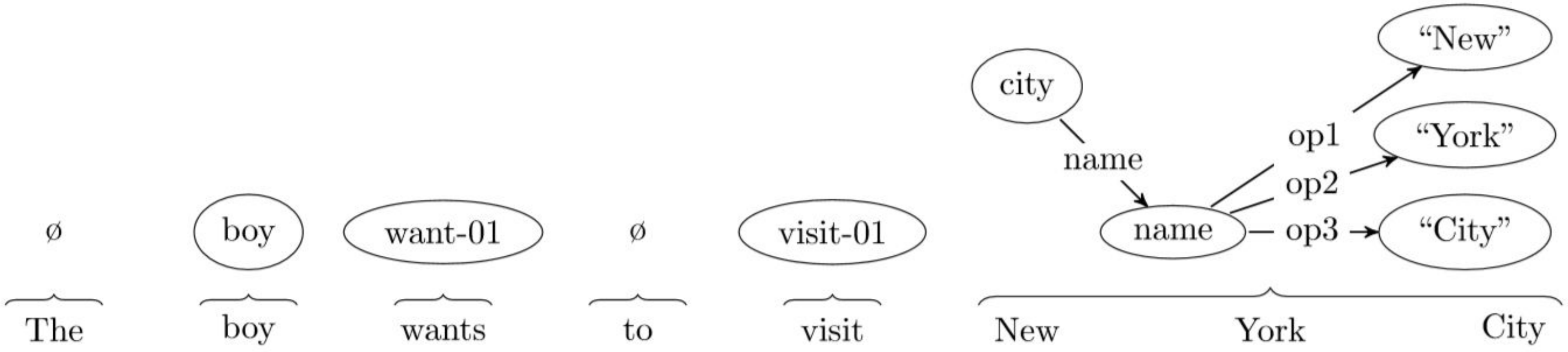
- Alignment in MT

e.g., German-to-English



- AMR Alignment

e.g., AMR-to-English



Alignments in AMR Parsing

Some AMR parsers rely on alignments: Composition-based parsers (e.g., Beschke, 2019; Lindemann et al., 2020; Groschwitz, 2019), transition-based parsers (Wang et al., 2015; Zhou et al., 2021; Astudillo et al., 2020; Naseem et al., 2019), factorization-based parsers (Flanigan et al., 2014)

For other AMR parsers (Lyu & Titov, 2018; Bevilacqua et al., 2021; Xu et al., 2020; Zhang et al., 2019), **explicit alignments could still be valuable for evaluation**.

Previous AMR Aligners

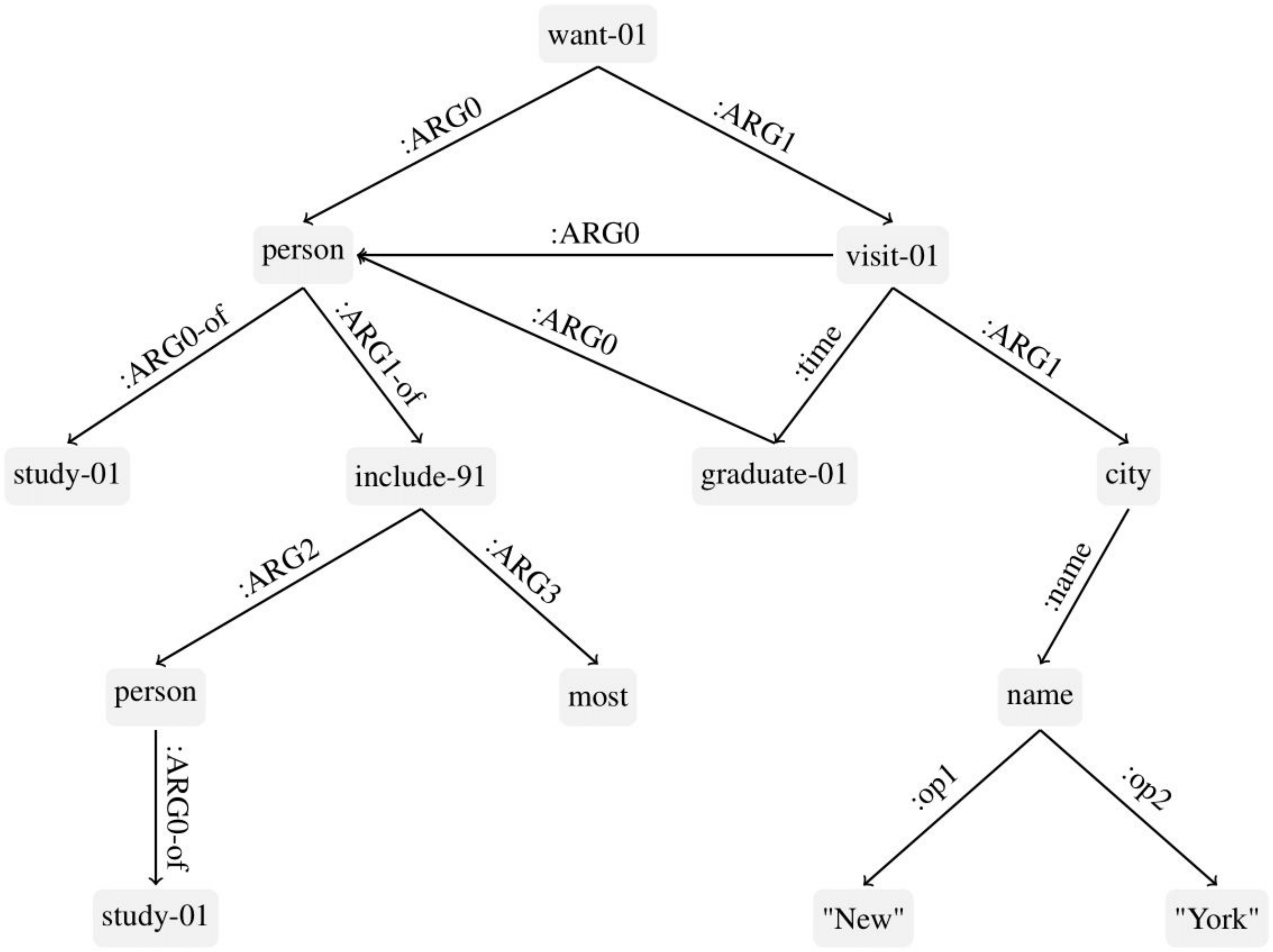
- **Rule Based:** JAMR alignments (Flanigan et al., 2014) align using iterative application of a list of rules.
- **Expectation-Maximization:** ISI alignments (Pourdamghani et al., 2014) first linearize an AMR and then apply an expectation-maximization alignment.
- **Tuned Alignments:** TAMR alignments (Liu et al., 2018) are built on top of the JAMR alignment system, but are tuned based on the performance of an oracle.
- **Graph Distance:** Wang and Xue (2017) use an HMM-based aligner and include a calculation of graph distance as a locality constraint, similar to our use of projection distance.

Limitations of Previous Aligners

Alignments are generally between individual nodes and individual tokens without full coverage:

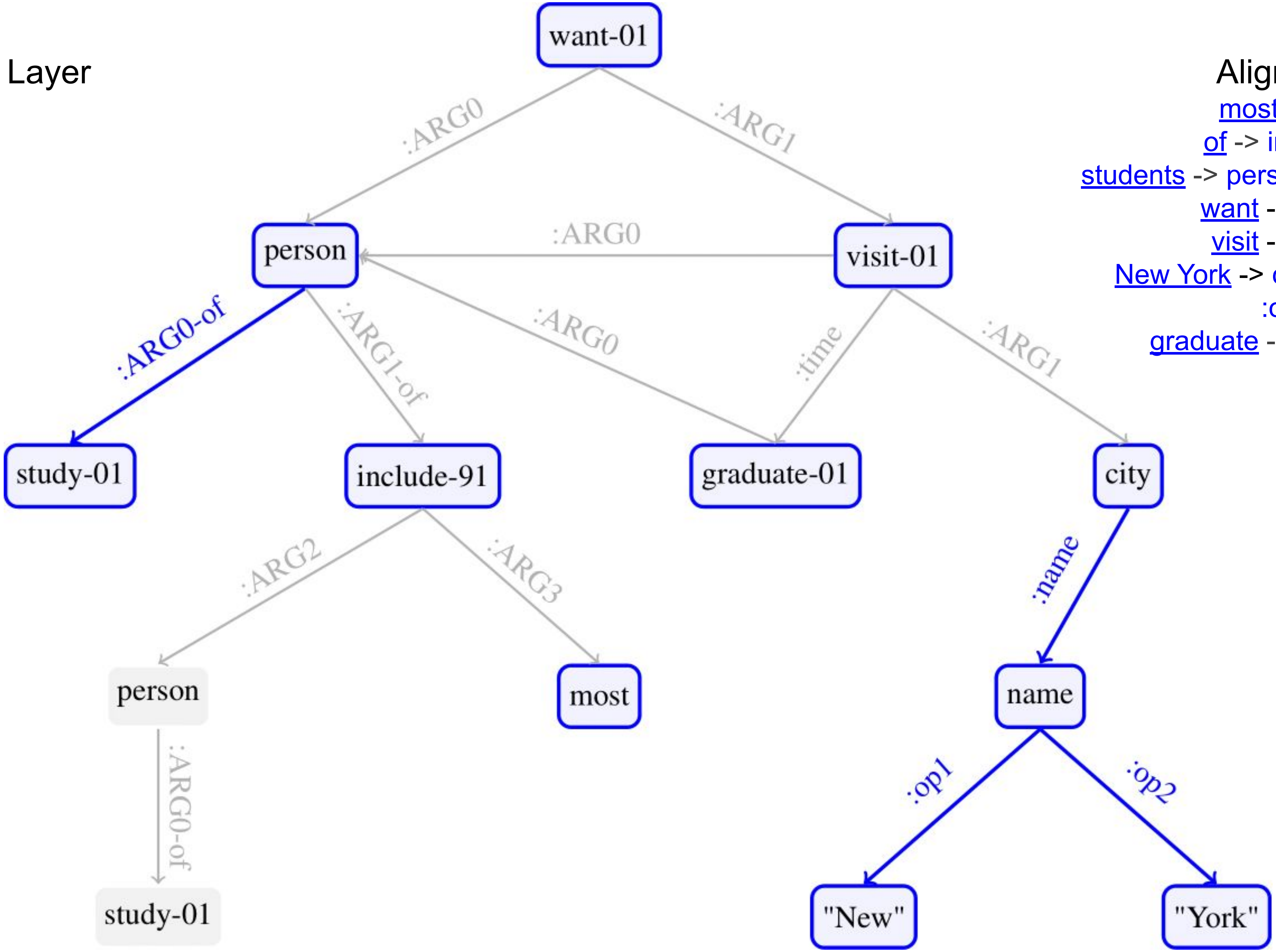
- Nodes in an alignment may be disconnected
- Lack of multi-token alignments
- Non-comprehensive node coverage
- Low coverage and performance on edges
- No alignment of reentrancies

	nodes	edges	reentrancies
JAMR	91.1	✗	✗
ISI	78.7	9.8	✗
TAMR	94.9	✗	✗



Most of the students want to visit New York when they graduate

Subgraph Layer



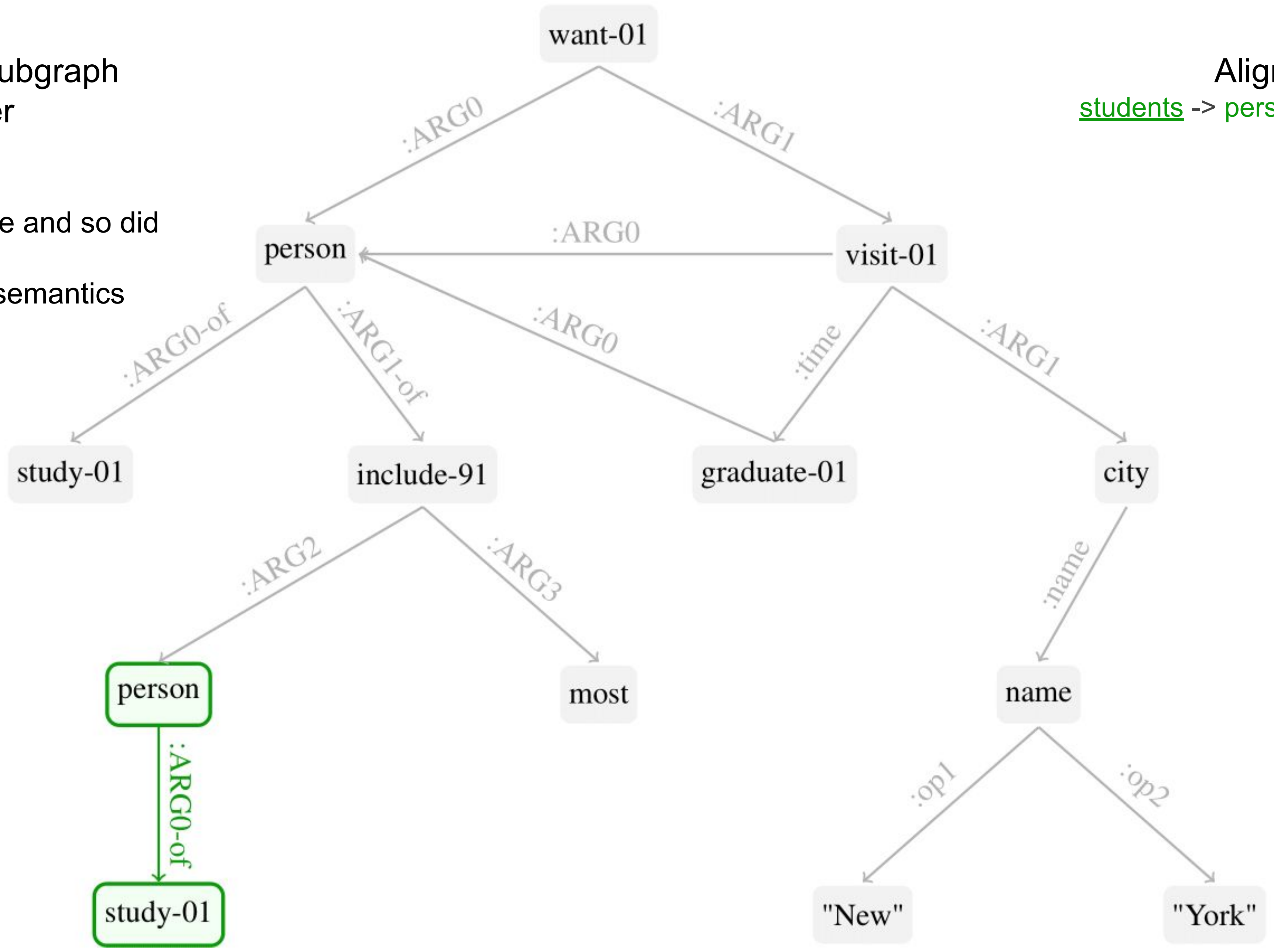
Alignments

most -> most
of -> include-91
students -> person :ARG0-of study-01
want -> want-01
visit -> visit-01
New York -> city :name (name :op1 “New” :op2 “York”)
graduate -> graduate-01

Most of the students want to visit New York when they graduate

Duplicate Subgraph Layer

used for:
ellipsis (“John ate and so did Mary”),
un-pronounced semantics

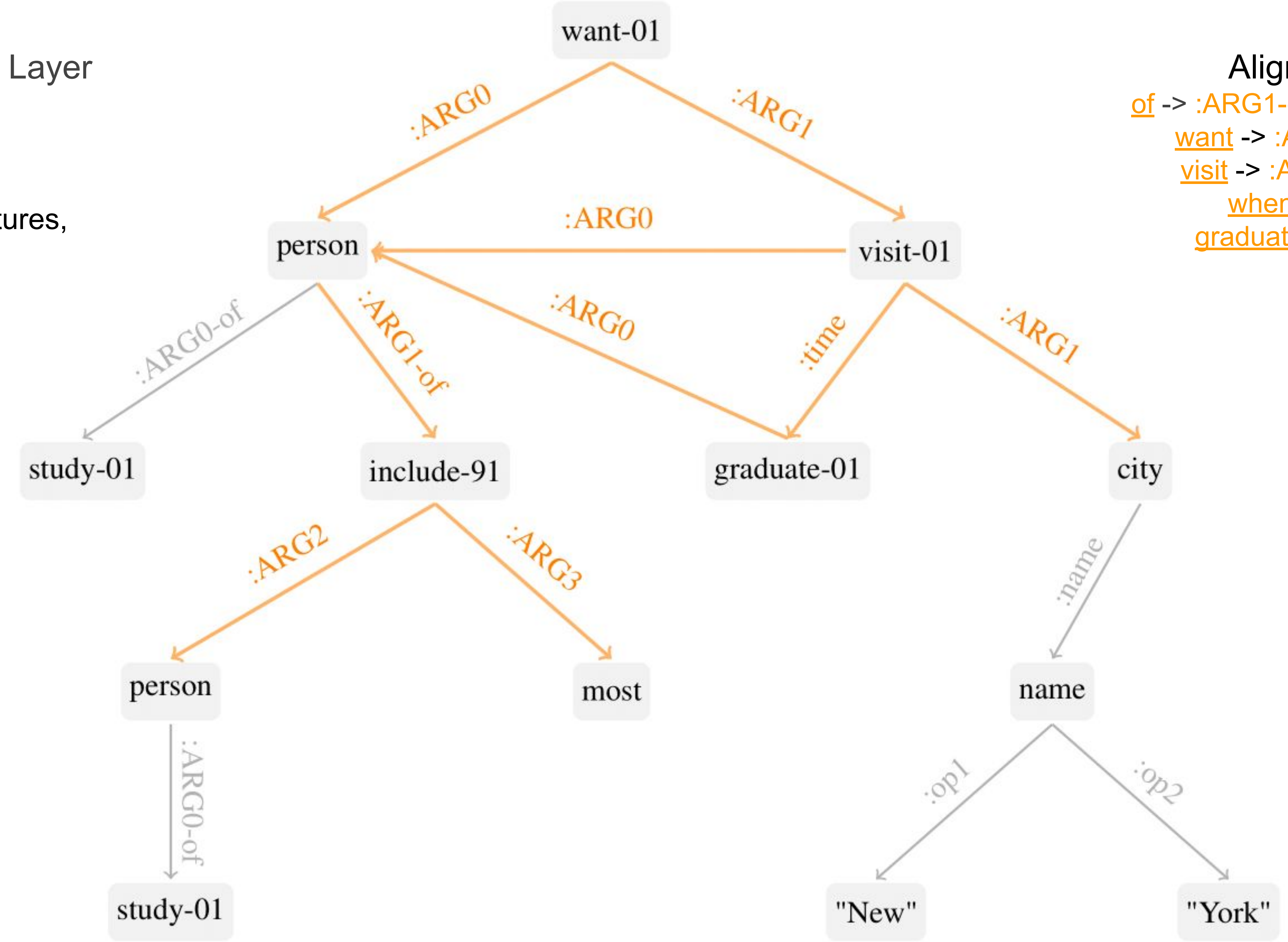


Alignments
students -> person :ARG0-of study-01

Most of the students want to visit New York when they graduate

Relation Layer

used for:
argument structures,
prepositions,
etc.



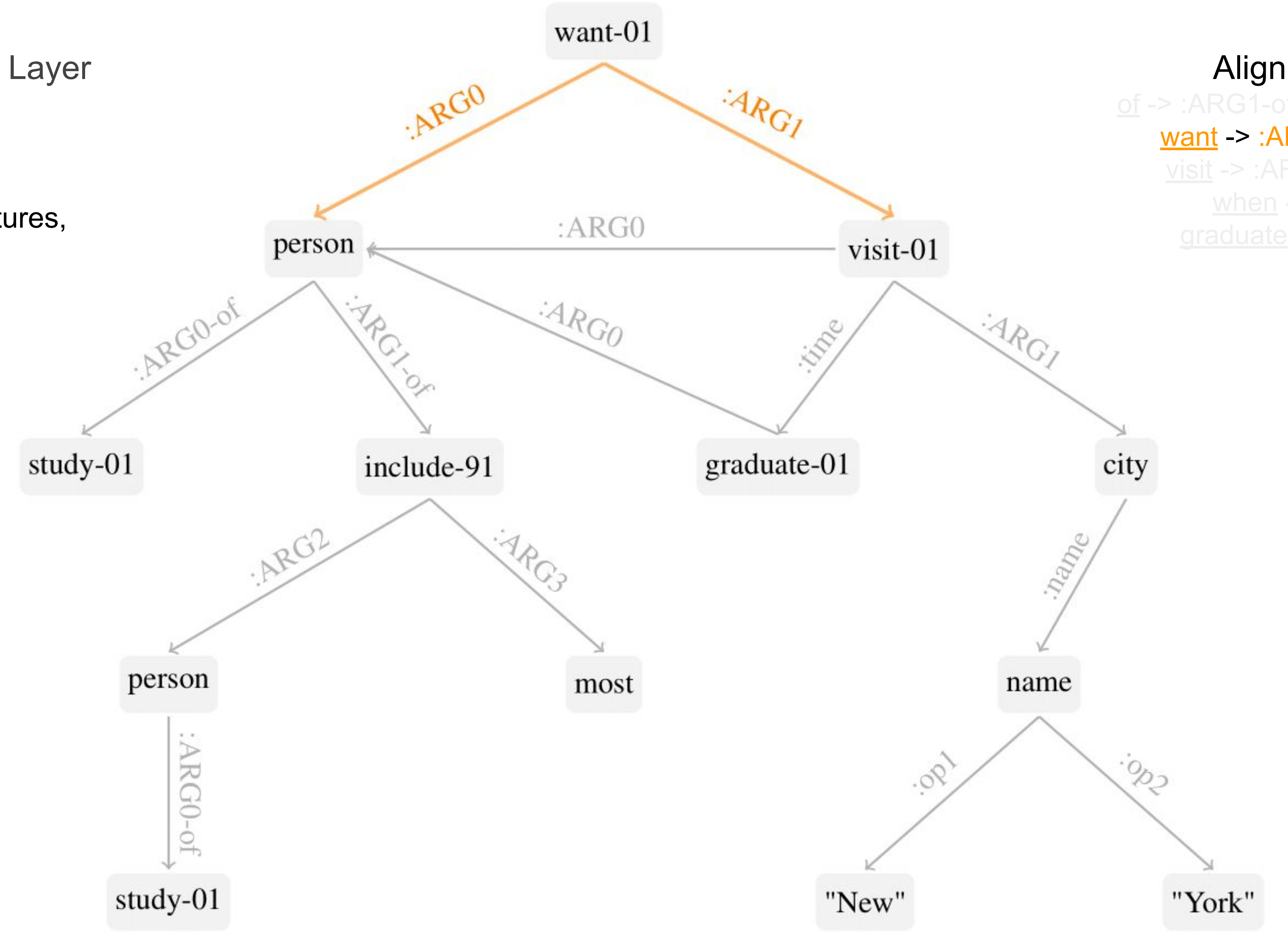
Alignments

of -> :ARG1-of :ARG2 :ARG3
want -> :ARG0 :ARG1
visit -> :ARG0 :ARG1
when -> :time
graduate -> :ARG0

Most of the students want to visit New York when they graduate

Relation Layer

used for:
argument structures,
prepositions,
etc.

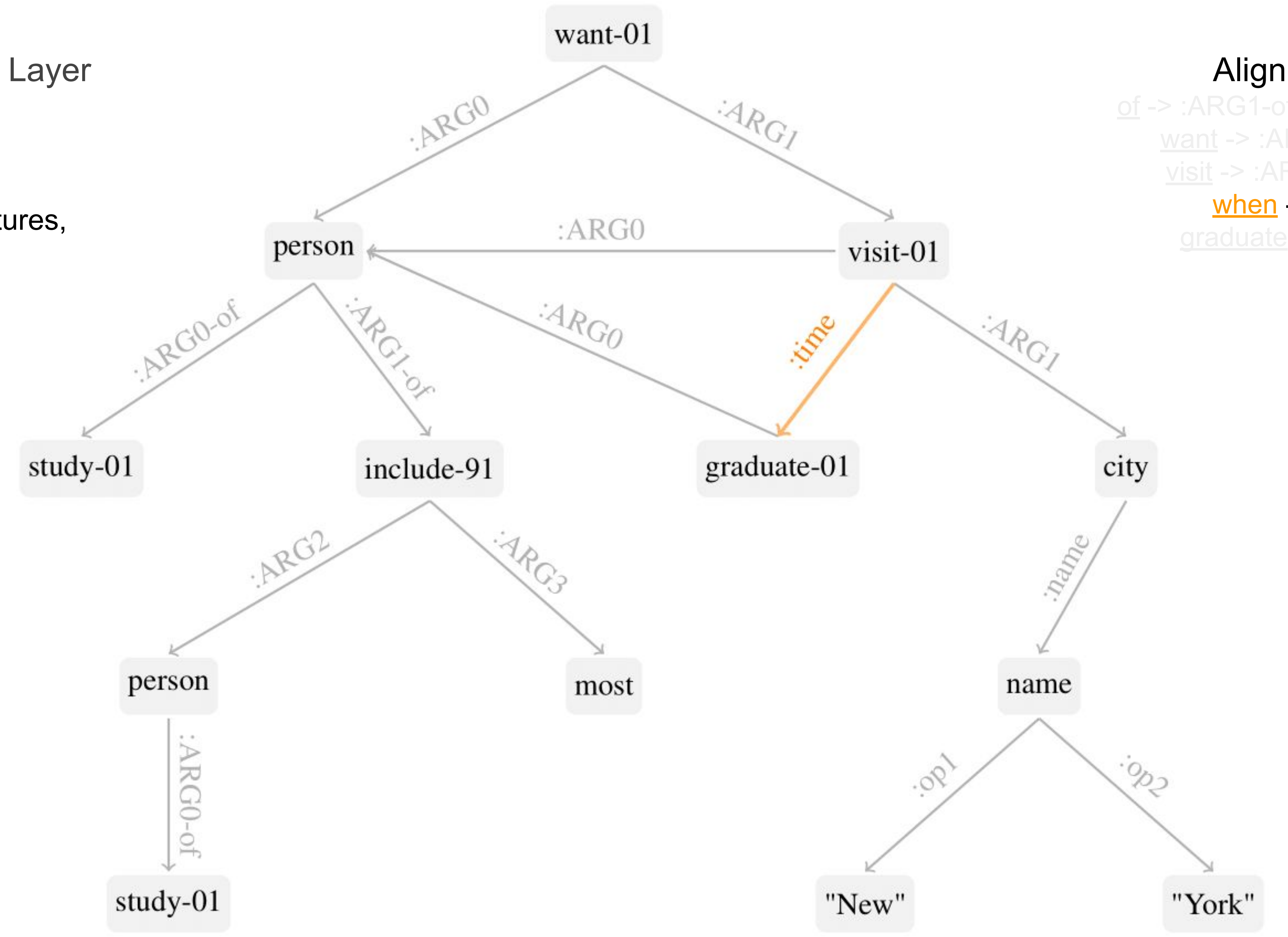


Alignments
of -> :ARG1-of :ARG2 :ARG3
want -> :ARG0 :ARG1
visit -> :ARG0 :ARG1
when -> :time
graduate -> :ARG0

Most of the students want to visit New York when they graduate

Relation Layer

used for:
argument structures,
prepositions,
etc.



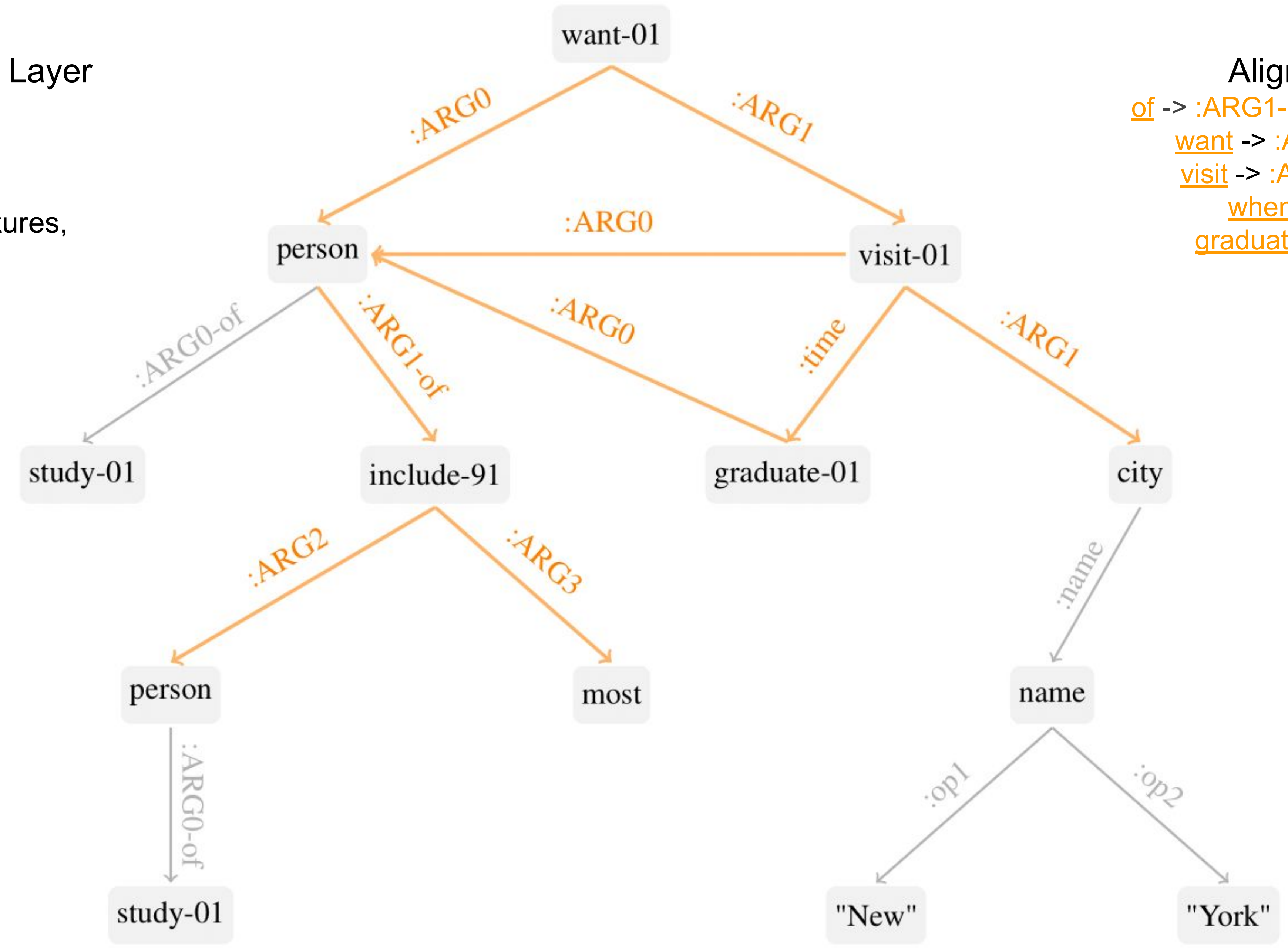
Alignments

of -> :ARG1-of :ARG2 :ARG3
want -> :ARG0 :ARG1
visit -> :ARG0 :ARG1
when -> :time
graduate -> :ARG0

Most of the students want to visit New York when they graduate

Relation Layer

used for:
argument structures,
prepositions,
etc.



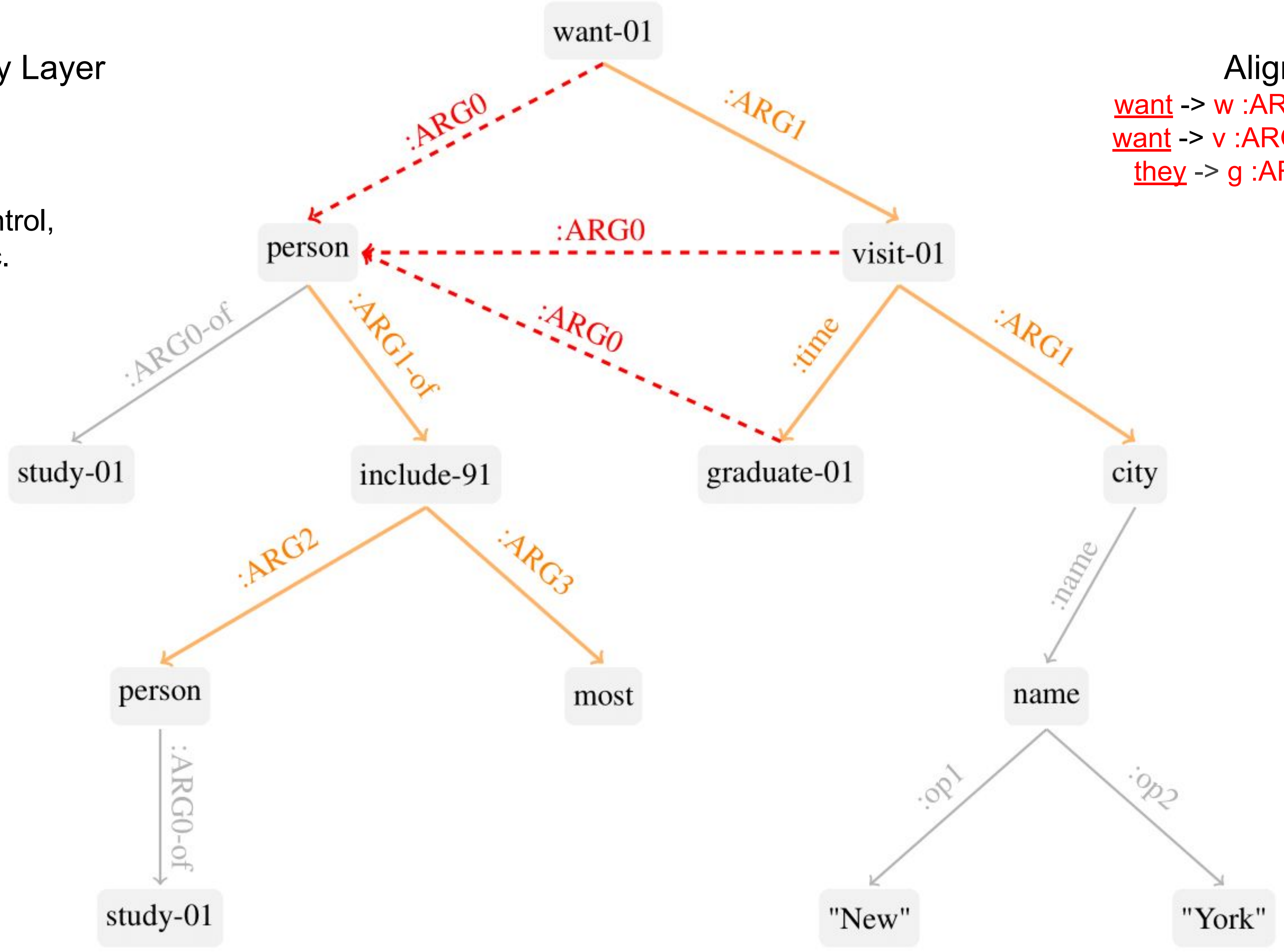
Alignments

of -> :ARG1-of :ARG2 :ARG3
want -> :ARG0 :ARG1
visit -> :ARG0 :ARG1
when -> :time
graduate -> :ARG0

Most of the students want to visit New York when they graduate

Reentrancy Layer

used for:
coreference, control, coordination, etc.



Alignments
want -> w :ARG0 p (PRIMARY)
want -> v :ARG0 p (CONTROL)
they -> g :ARG0 p (COREF)

Most of the students want to visit New York when they graduate

LEAMR Released Data

Automatic Alignments:

- AMR Release 3.0
- Little Prince

Gold Alignments:

- 350 sentences

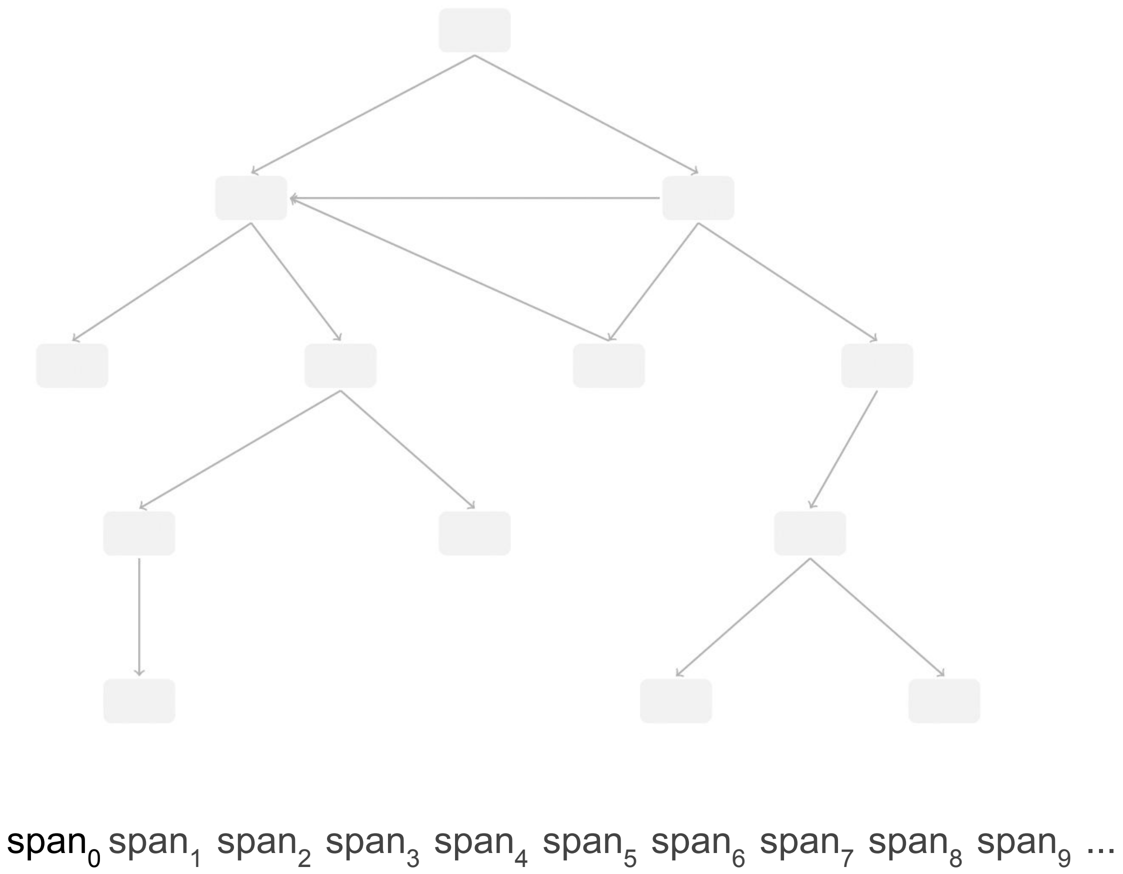
IAA	Exact Align F1
Subgraphs (366)	94.54
Relations (260)	90.73
Reentrancies (65)	76.92
Duplicates (5)	66.67

https://github.com/ablodge/leamr

Structure-Aware EM Algorithm

For alignable elements (unaligned nodes or edges) of the graph, do until finished:

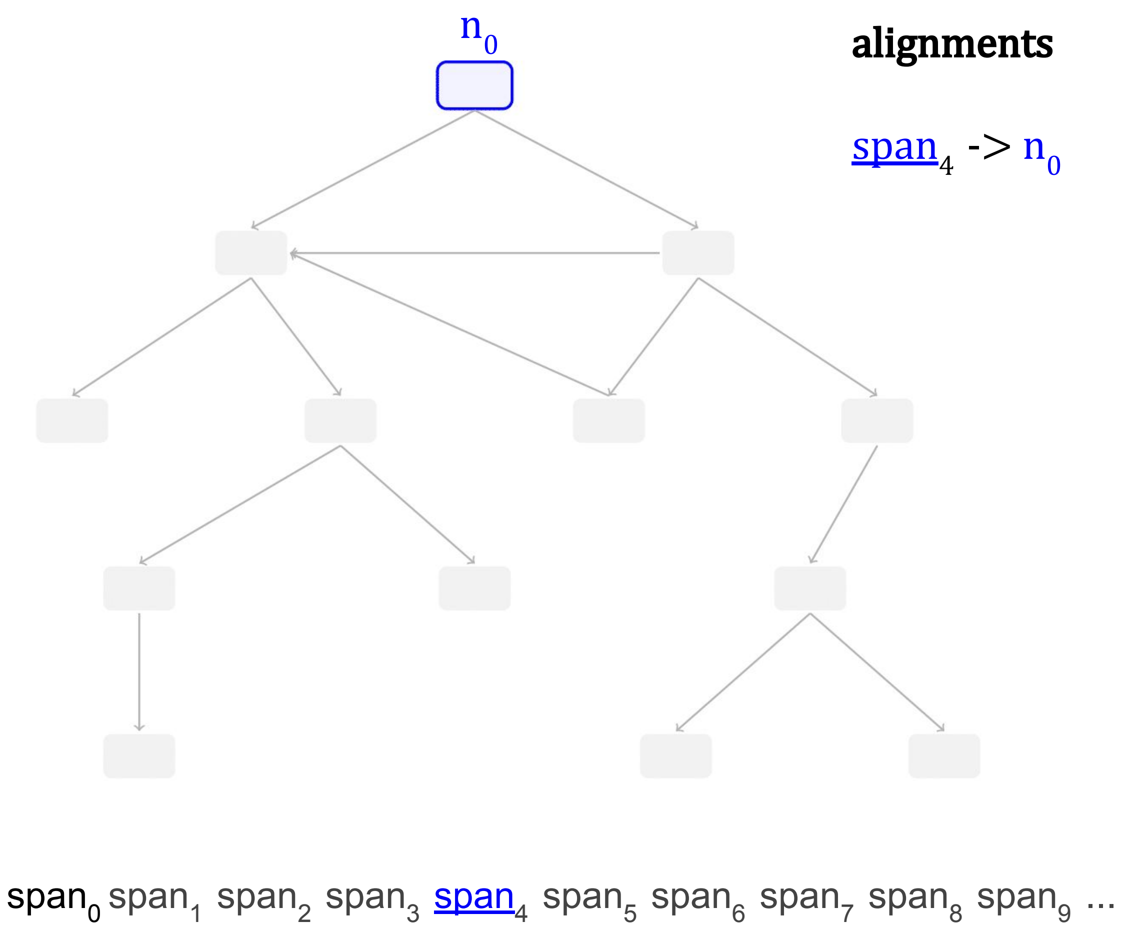
- Identify legal candidate spans
  - unaligned spans
  - spans aligned to a neighboring element
  - (for subgraphs only) any span aligned to a *duplicate* of this element
- Score each candidate based on alignment and distance probabilities
- Align best scoring element-span pair



Structure-Aware EM Algorithm

For alignable elements (unaligned nodes or edges) of the graph, do until finished:

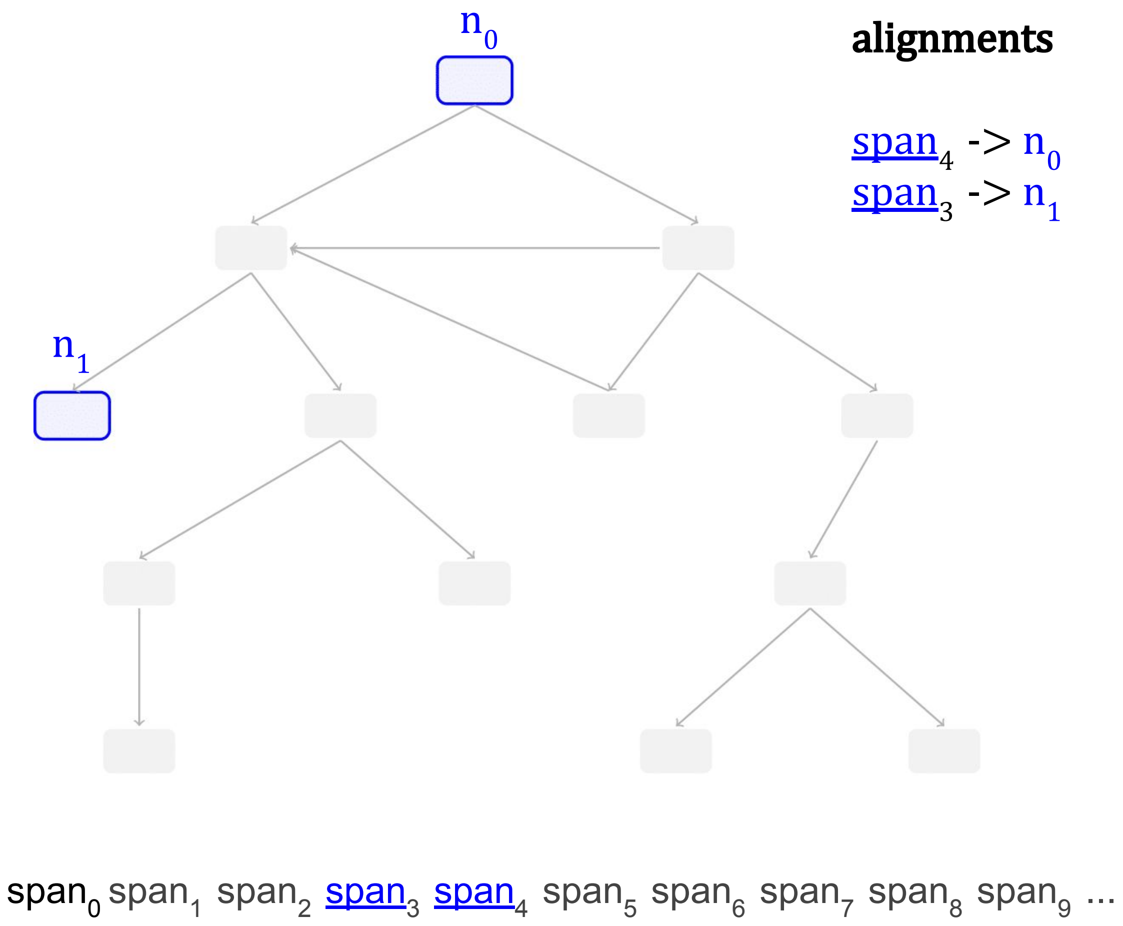
- Identify legal candidate spans
  - unaligned spans
  - spans aligned to a neighboring element
  - (for subgraphs only) any span aligned to a *duplicate* of this element
- Score each candidate based on alignment and distance probabilities
- Align best scoring element-span pair



Structure-Aware EM Algorithm

For alignable elements (unaligned nodes or edges) of the graph, do until finished:

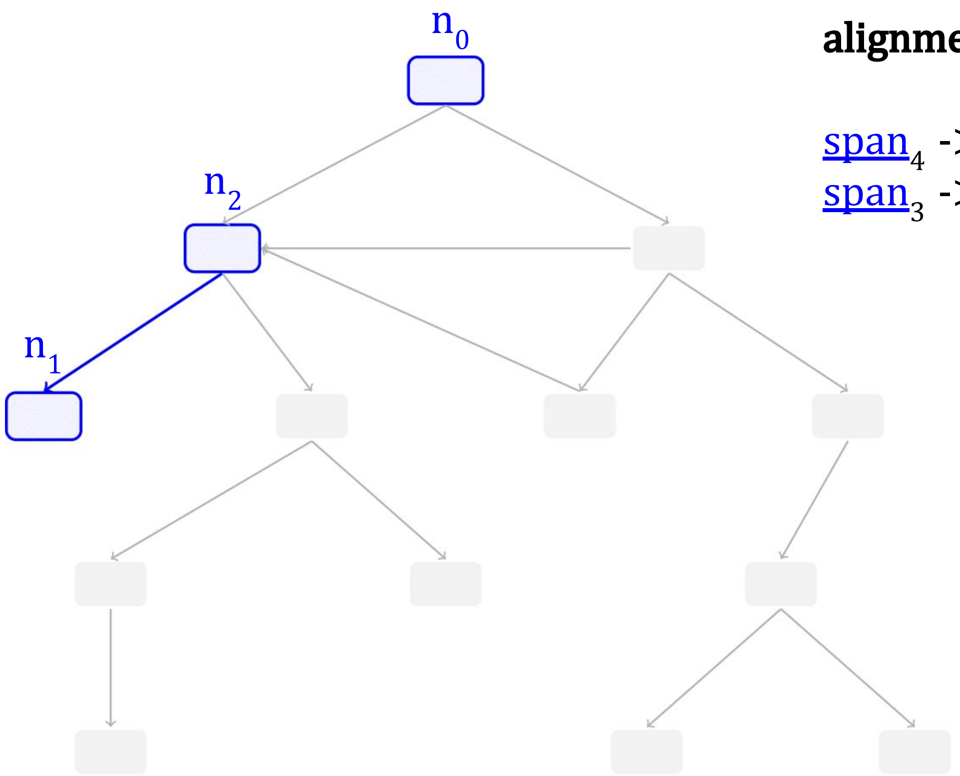
- Identify legal candidate spans
  - unaligned spans
  - spans aligned to a neighboring element
  - (for subgraphs only) any span aligned to a *duplicate* of this element
- Score each candidate based on alignment and distance probabilities
- Align best scoring element-span pair



Structure-Aware EM Algorithm

For alignable elements (unaligned nodes or edges) of the graph, do until finished:

- Identify legal candidate spans
  - unaligned spans
  - spans aligned to a neighboring element
  - (for subgraphs only) any span aligned to a *duplicate* of this element
- Score each candidate based on alignment and distance probabilities
- Align best scoring element-span pair



alignments

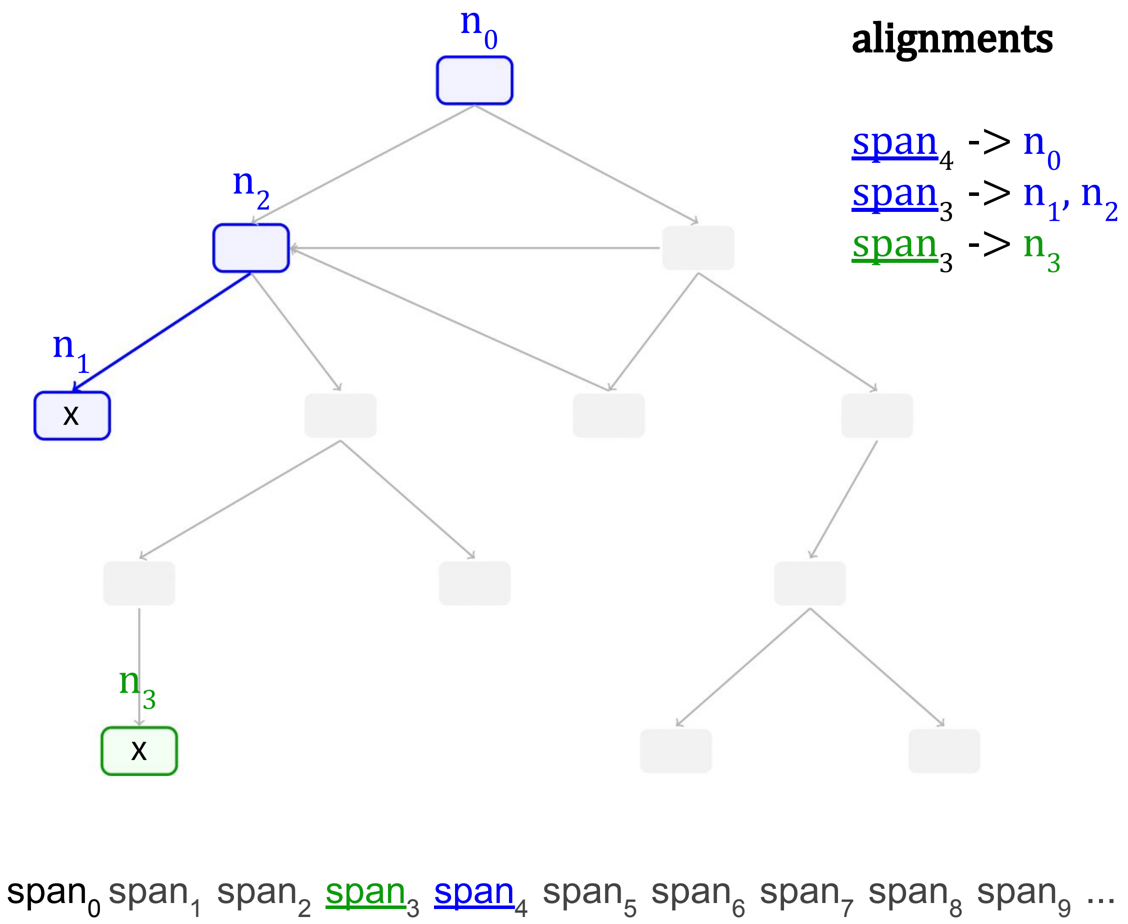
$span_4$ -> n_0
$span_3$ -> n_1, n_2

$span_0$ $span_1$ $span_2$ $span_3$ $span_4$ $span_5$ $span_6$ $span_7$ $span_8$ $span_9$ ...

Structure-Aware EM Algorithm

For alignable elements (unaligned nodes or edges) of the graph, do until finished:

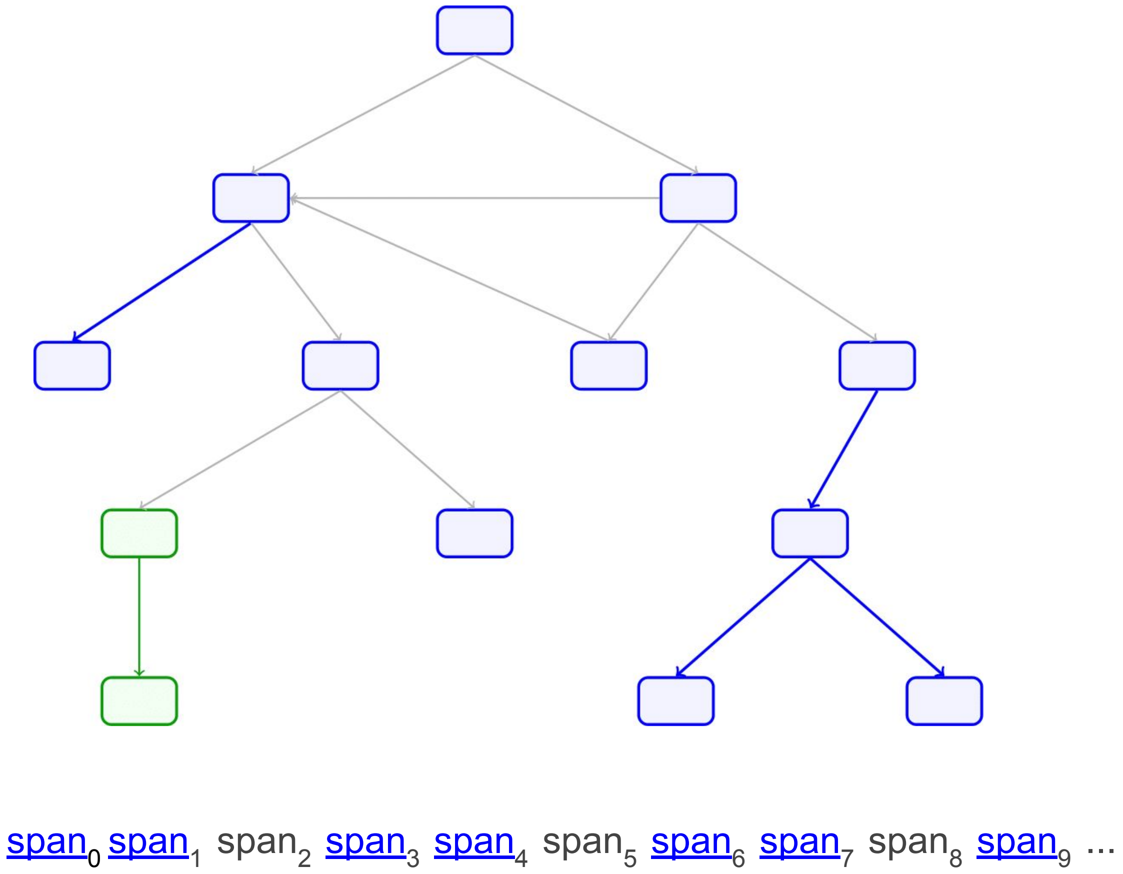
- Identify legal candidate spans
    - unaligned spans
    - spans aligned to a neighboring element
    - (for subgraphs only) any span aligned to a *duplicate* of this element
- Score each candidate based on alignment and distance probabilities
- Align best scoring element-span pair



Structure-Aware EM Algorithm

For alignable elements (unaligned nodes or edges) of the graph, do until finished:

- Identify legal candidate spans
  - unaligned spans
  - spans aligned to a neighboring element
  - (for subgraphs only) any span aligned to a *duplicate* of this element
- Score each candidate based on alignment and distance probabilities
- Align best scoring element-span pair

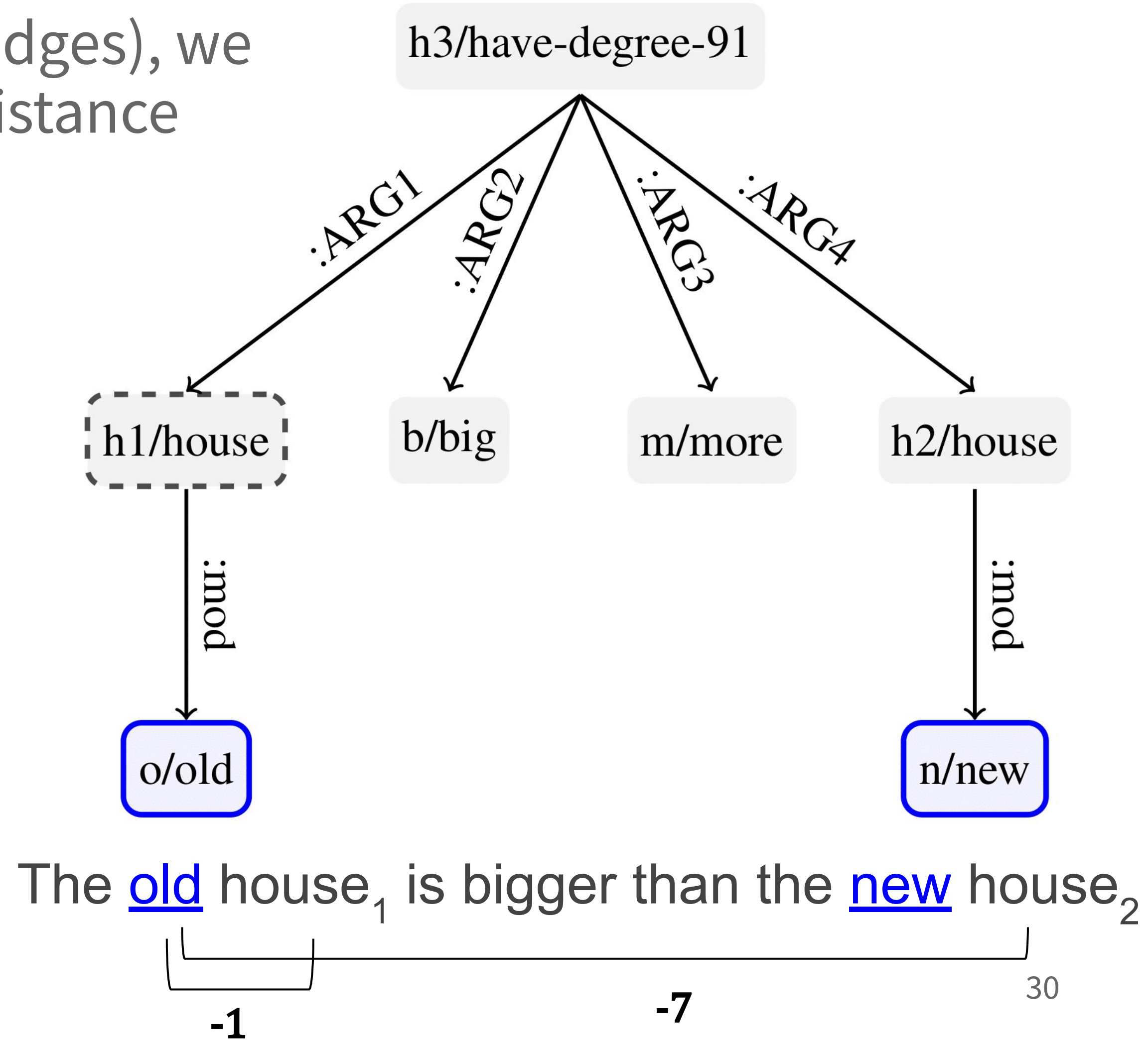


Projection Distance

alignments

old -> old
new -> new

For two neighboring elements (nodes or edges), we define projection distance as the *signed* distance between spans aligned to each element.

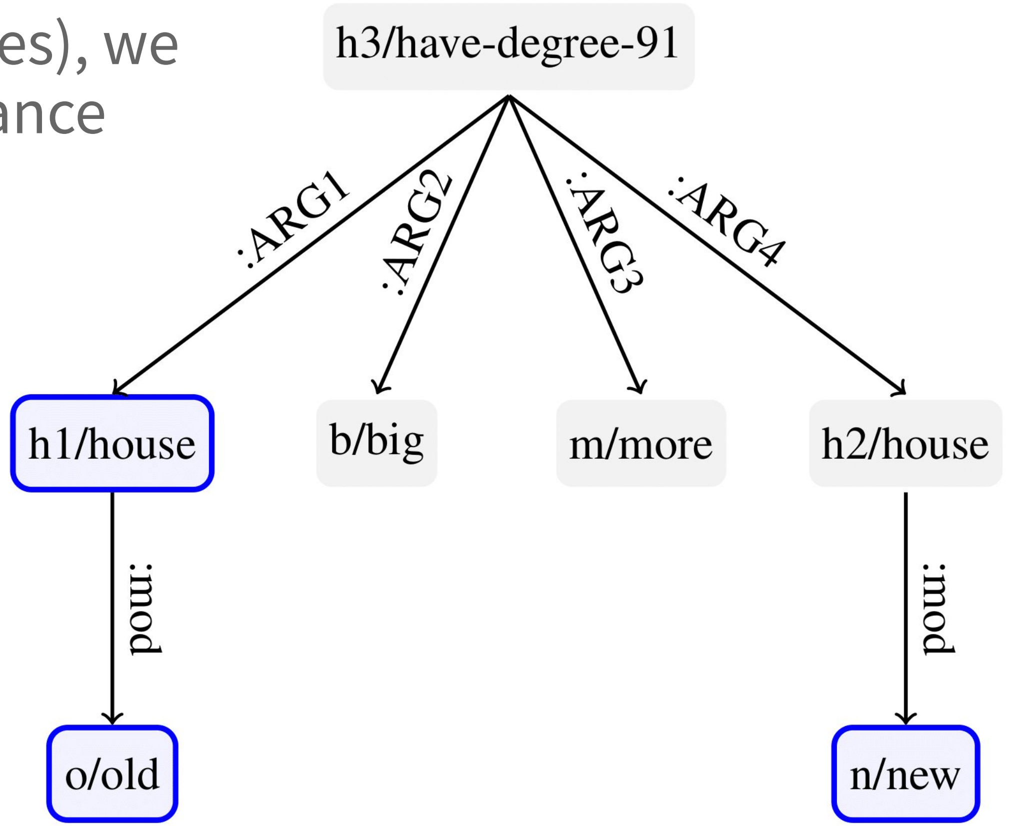


Projection Distance

alignments

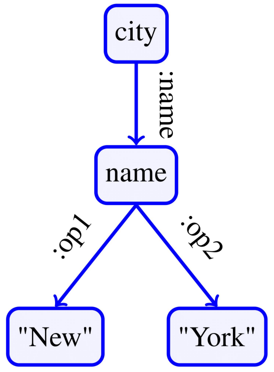
old -> old
new -> new
house$_1$ -> h1/house

For two neighboring elements (nodes or edges), we define projection distance as the *signed* distance between spans aligned to each element.



The old house$_1$ is bigger than the new house$_2$

Aligning Subgraphs



New York

$$\text{score}(\langle g, s \rangle) = \underbrace{P_{\text{align}}(g \mid s; \theta_1)}_{\text{subgraph label given span label}} \cdot \underbrace{\prod_{d_i \in D} P_{\text{dist}}(d_i; \theta_2)^{\frac{1}{|D|}}}_{\text{projection distance probability}} \cdot \underbrace{IB(g, s)}_{\text{inductive bias}}$$

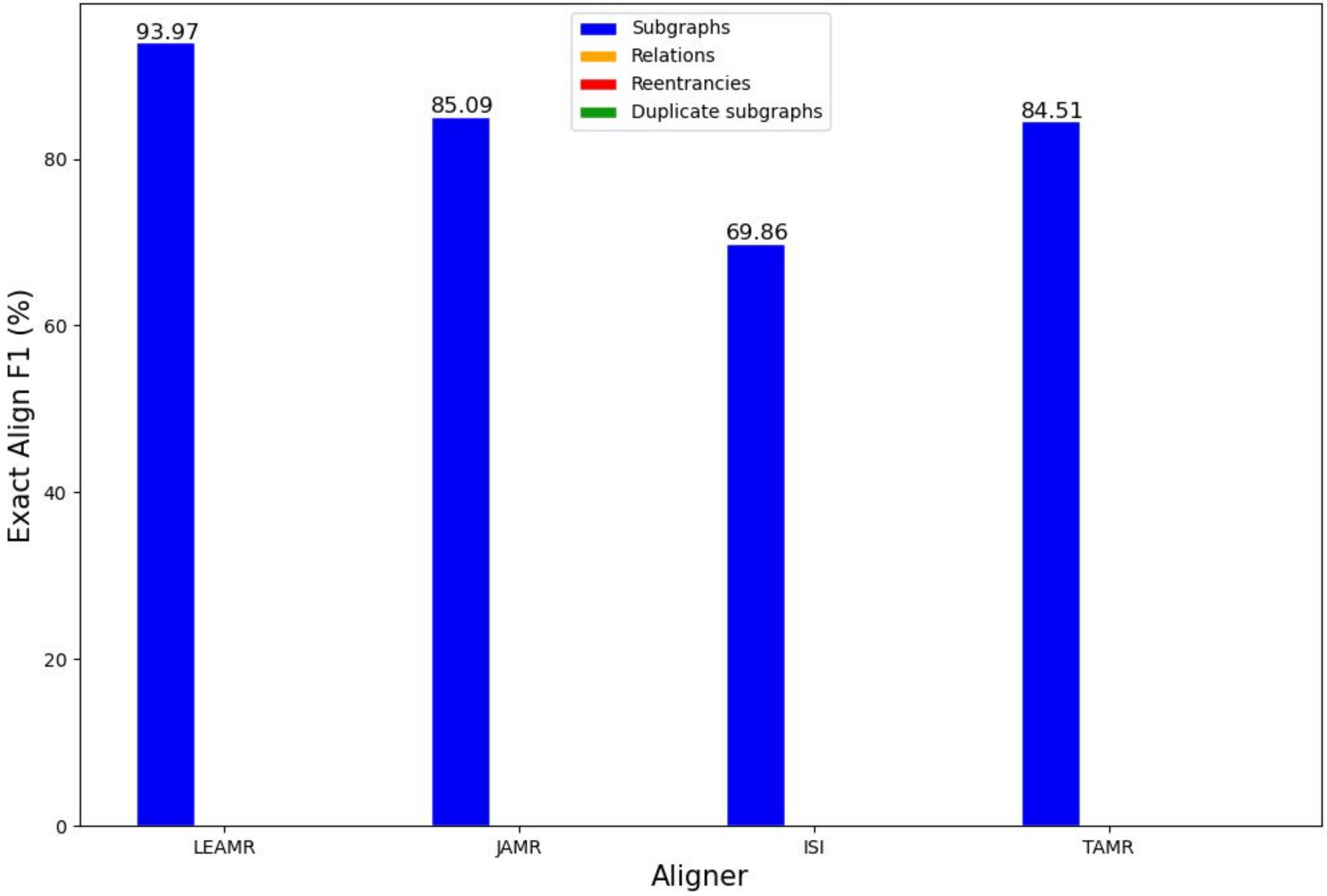
Aligning Relations

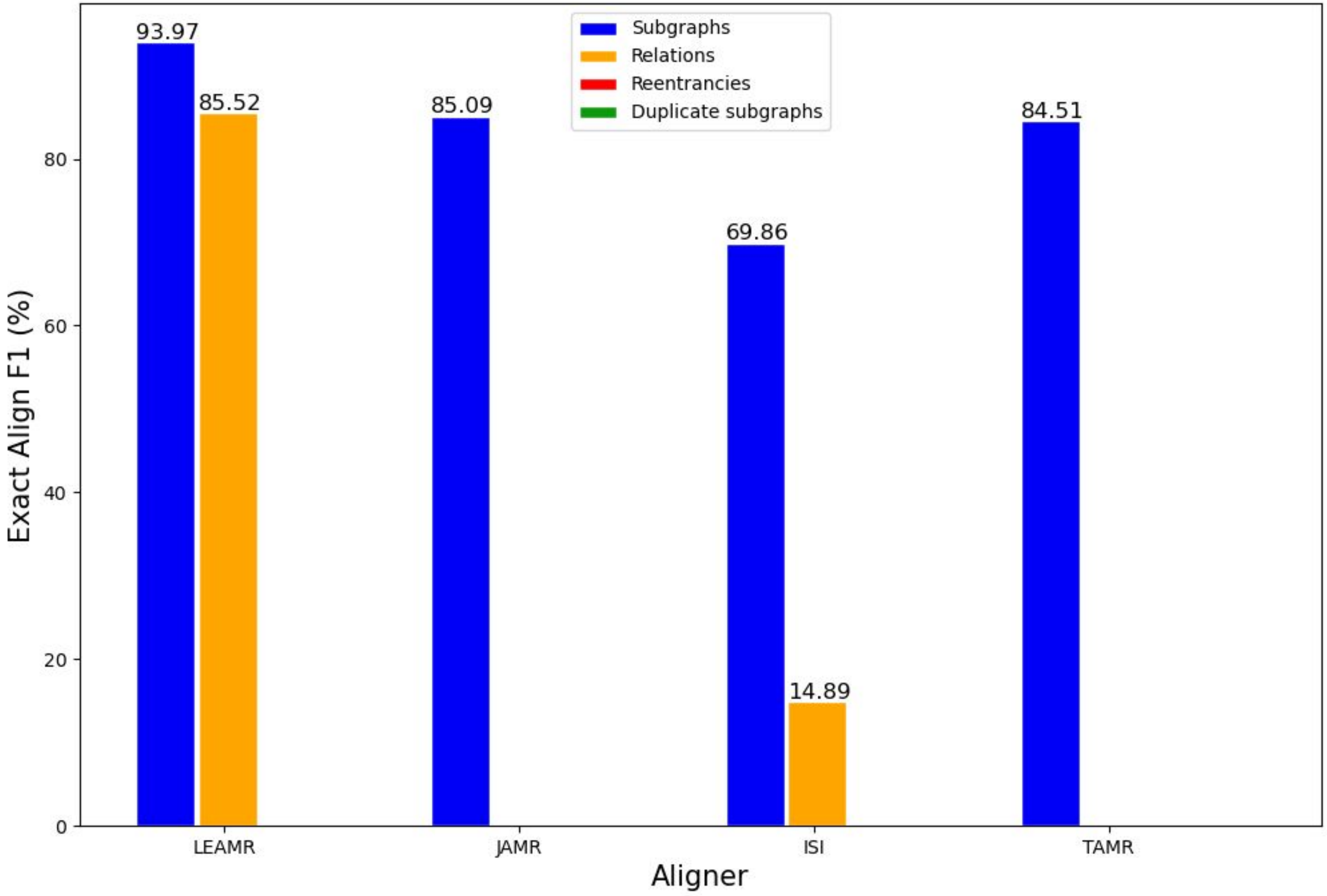
$\xrightarrow{\text{:time}}$ when

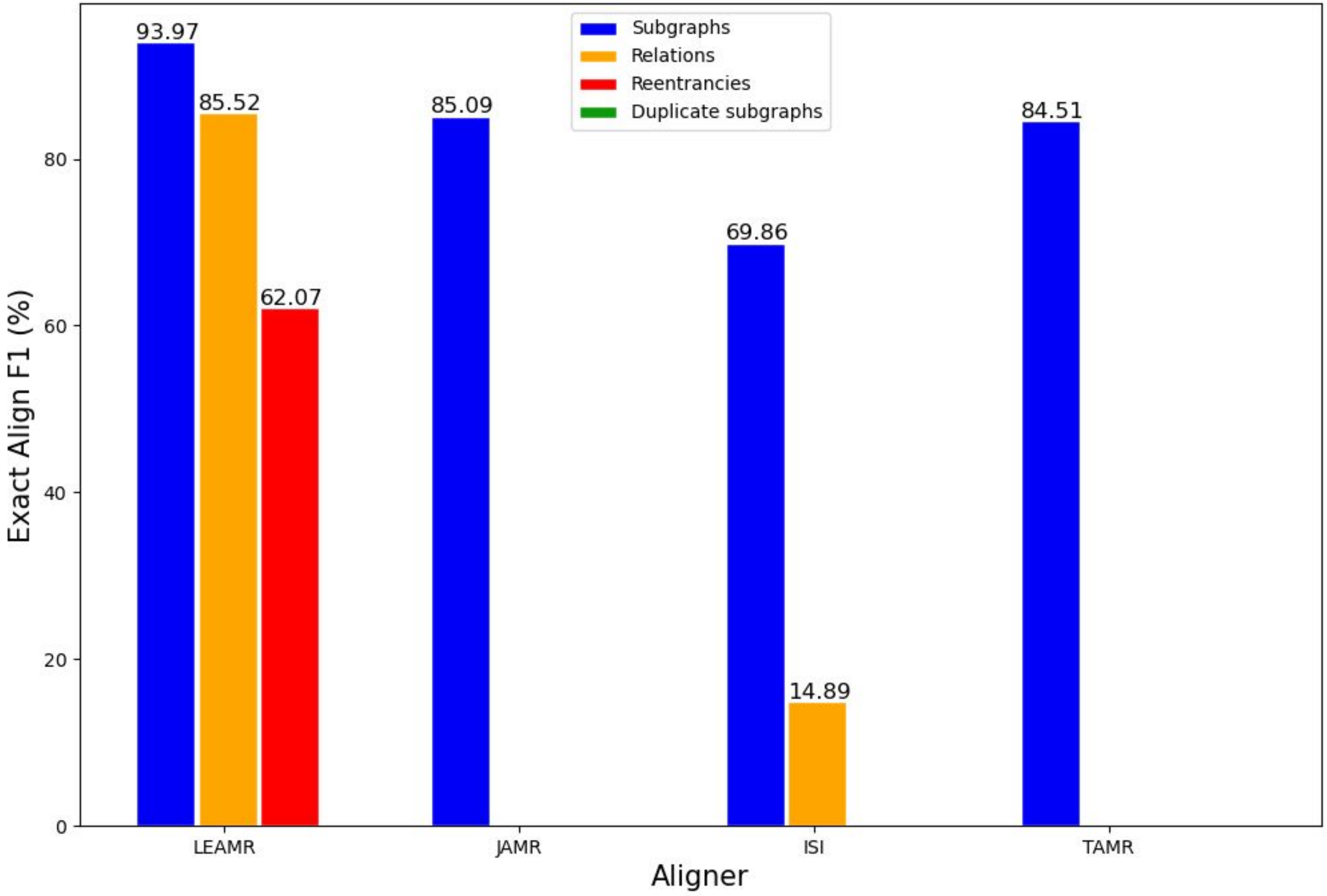
$$score(\langle a, s \rangle) = \underbrace{P_{\text{align}}(a \mid s; \theta_3)}_{\text{reentrancy label given span label}} \cdot \underbrace{\prod_{d_i \in D_1} P_{\text{dist}}(d_i; \theta_4)^{\frac{1}{|D_1|}}}_{\text{projection distance probability (parent)}} \cdot \underbrace{\prod_{d_j \in D_2} P_{\text{dist}}(d_j; \theta_5)^{\frac{1}{|D_2|}}}_{\text{projection distance probability (child)}}$$

Aligning Reentrancies

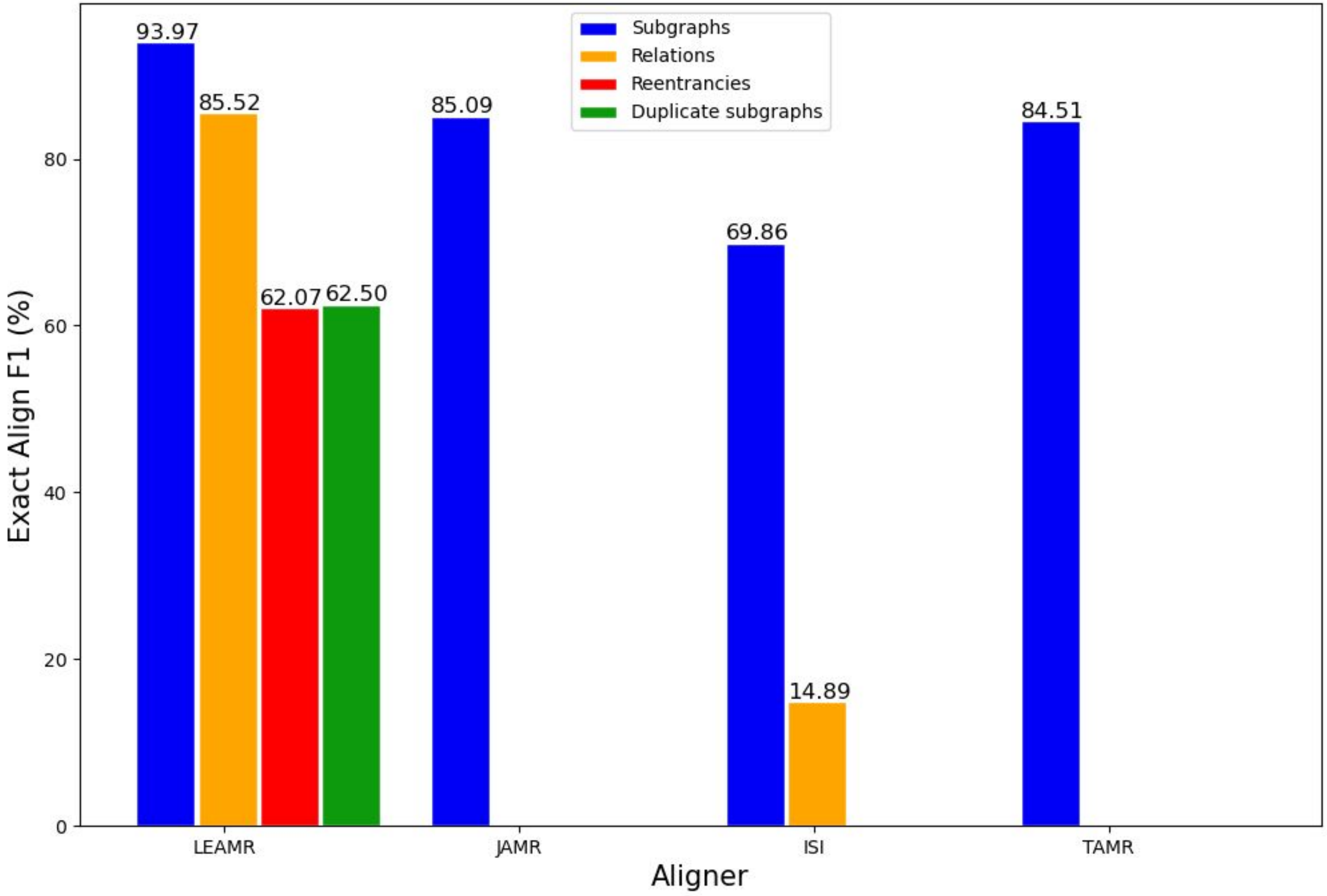
$$score(\langle r, s, type \rangle) = \underbrace{P_{\text{align}}(\overset{\xrightarrow{\text{:ARG0}}}{r}, \overset{\text{COREF}}{type} \mid \overset{\underline{\text{they}}}{s}; \theta_6)}_{\text{reentrancy label given span label}} \cdot \underbrace{P_{\text{dist}}(d_1; \theta_7)}_{\text{projection distance probability (parent)}} \cdot \underbrace{P_{\text{dist}}(d_2; \theta_8)}_{\text{projection distance probability (child)}}$$


Subgraphs
Relations
Reentrancies
Duplicate subgraphs
93.97
85.09
69.86
84.51
Exact Align F1 (%)
0
20
40
60
80
LEAMR
JAMR
ISI
TAMR
Aligner


Subgraphs
Relations
Reentrancies
Duplicate subgraphs
93.97
85.52
85.09
69.86
14.89
84.51
Exact Align F1 (%)
0
20
40
60
80
LEAMR
JAMR
ISI
TAMR
Aligner



Subgraphs
Relations
Reentrancies
Duplicate subgraphs
93.97
85.52
62.07
85.09
69.86
14.89
84.51
0
20
40
60
80
Exact Align F1 (%)
LEAMR
JAMR
ISI
TAMR
Aligner


93.97
85.52
62.07
62.50
85.09
69.86
14.89
84.51
Subgraphs
Relations
Reentrancies
Duplicate subgraphs
Exact Align F1 (%)
0
20
40
60
80
LEAMR
JAMR
ISI
TAMR
Aligner

DATA + CODE:

https://github.com/ablodge/leamr

Other AMR research:
https://nert-nlp.github.io/AMR-Bibliography/

Thank You!