Part II:
Algorithms and Applications

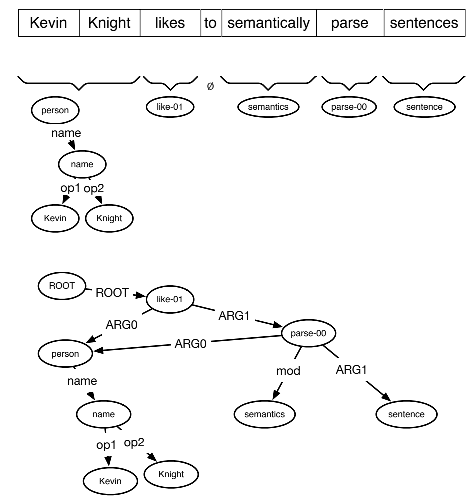
Speaker: Jeffrey Flanigan

Alignment

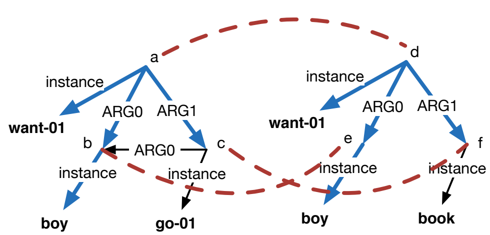
IAEA accepted North Korea 's proposal in November.

```
(a / accept-01
     :ARG0 (o / organization
          :ARG0 (n / name
               :op1 "IAEA"))
     :ARG1 (t2 / thing
          :ARG1-of (p / propose-01
               :ARG0 (c / country
                    :name (n2 / name
                         :op1 "North"
                         :op2 "Korea"))))
     :time (d / date-entity
               :month 11))
```

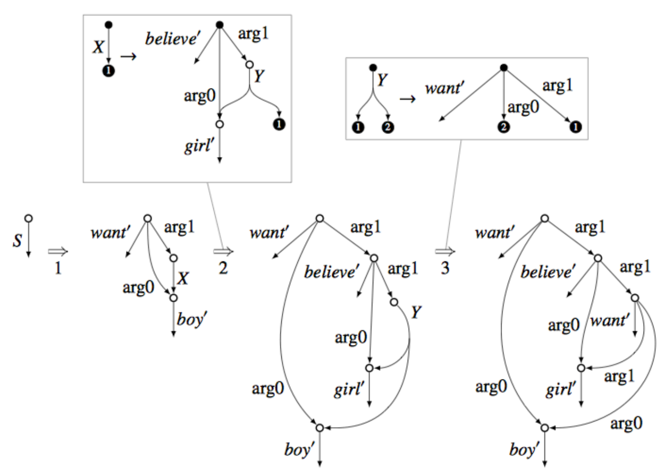
Parsing



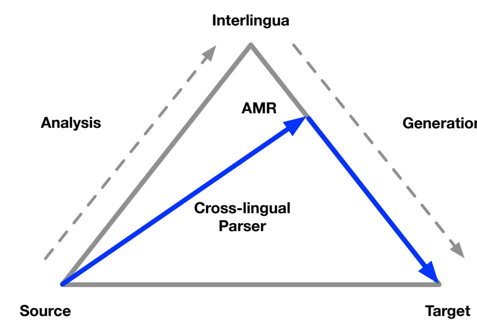
Evaluation



Graph grammars



Applications



Outline

- Alignment
- Parsing
- Evaluation
- Graph Grammars and Automata
- Applications

Alignment: Motivation

- AMR annotation has no explicit alignment to sentence
- Training data has whole sentence – AMR graph pairs
- For generalization performance, need fine-grained correspondence between words and pieces of AMR
- Alignments provide this correspondence

Need alignments to train parsers, etc

Alignment

The tour was a surprise offer made *by* North Korea *in* November.

```
(t / thing
      :ARG0-of (s / surprise-01)
      :ARG1-of (o / offer-01
            :ARG0 (c / country
                  :name (n / name
                        :op1 "North"
                        :op2 "Korea"))
            :time (d / date-entity
                        :month 11))
      :domain (t2 / tour-01))
```

- Align concepts with words
- Can also align edges with function words

Alignment

- Alignment
  - Motivation
  - **JAMR's rule-based aligner**
  - Alignment w/ EM
- Parsing
- Graph Formalisms
- Applications

JAMR Aligner (Flanigan et al, 2014)

- Aligns graph fragments to spans of words (edges not in fragments are unaligned)
- Uses a set of handcrafted rules
- Uses lemmatizer, string edit distance to match concepts with words
- Rules for: named entities, date entities, special concepts, negation, degrees, etc (15 total rules)

JAMR Aligner

For each rule

- Greedily align concepts in a depth first traversal of the AMR graph
- Rules are applied in a specified order

JAMR Aligner

IAEA accepted North Korea 's proposal in November.

```
(a / accept-01
    :ARG0 (o / organization
        :ARG0 (n / name
            :op1 "IAEA"))
    :ARG1 (t2 / thing
        :ARG1-of (p / propose-01
            :ARG0 (c / country
                :name (n2 / name
                    :op1 "North"
                    :op2 "Korea"))))
    :time (d / date-entity
            :month 11))
```

JAMR Aligner

Rule 1) Date entity

IAEA accepted North Korea 's proposal in November.

```
(a / accept-01
     :ARG0 (o / organization
          :ARG0 (n / name
               :op1 "IAEA"))
     :ARG1 (t2 / thing
          :ARG1-of (p / propose-01
               :ARG0 (c / country
                    :name (n2 / name
                         :op1 "North"
                         :op2 "Korea"))))
     :time (d / date-entity
               :month 11))
```

JAMR Aligner

Rule 3) Named entity

IAEA accepted North Korea 's proposal in November.

```
(a / accept-01
    :ARG0 (o / organization
        :ARG0 (n / name
            :op1 "IAEA"))
    :ARG1 (t2 / thing
        :ARG1-of (p / propose-01
            :ARG0 (c / country
                :name (n2 / name
                    :op1 "North"
                    :op2 "Korea"))))
    :time (d / date-entity
            :month 11))
```

JAMR Aligner

Rule 5) Single concept

IAEA accepted North Korea 's proposal in November.

```
(a / accept-01
     :ARG0 (o / organization
          :ARG0 (n / name
               :op1 "IAEA"))
     :ARG1 (t2 / thing
          :ARG1-of (p / propose-01
               :ARG0 (c / country
                    :name (n2 / name
                         :op1 "North"
                         :op2 "Korea"))))
     :time (d / date-entity
               :month 11))
```

JAMR Aligner

Rule 10) person-of/thing-of

IAEA accepted North Korea 's proposal in November.

```
(a / accept-01
     :ARG0 (o / organization
          :ARG0 (n / name
               :op1 “IAEA”))
     :ARG1 (t2 / thing
          :ARG1-of (p / propose-01
               :ARG0 (c / country
                    :name (n2 / name
                         :op1 “North”
                         :op2 “Korea”))))
     :time (d / date-entity
               :month 11))
```

JAMR Aligner

IAEA accepted North Korea 's proposal in November.

```
(a / accept-01
     :ARG0 (o / organization
          :ARG0 (n / name
               :op1 "IAEA"))
     :ARG1 (t2 / thing
          :ARG1-of (p / propose-01
               :ARG0 (c / country
                    :name (n2 / name
                         :op1 "North"
                         :op2 "Korea"))))
     :time (d / date-entity
               :month 11))
```

JAMR Aligner

Evaluate on 200 hand-aligned sentences:

F_1: 90%   Precision: 92%   Recall: 89%

<u>Extracted concept table</u>

```
8       critical => (critical)
2       critical => (criticize-01)
1       critically => (critical)
1       criticised => (criticize-01)
14      criticism => (criticize-01)
30      criticized => (criticize-01)
1       critics => (critic)
4       critics => (person :ARG0-of (criticize-01))
5       crop => (crop)
5       crops => (crop)
3       cross => (cross)
15      cross => (cross-02)
3       cross => (cross-border)
2       cross => (cross-strait)
1       crossed => (cross-00)
2       crossing => (cross-02)
```

ISI Aligner (Pourdamghani et al, 2014)

- Aligns single concept or edge to single word
- Learns from data using EM
- Inspired by MT alignment models

ISI Aligner

IAEA accepted North Korea 's proposal in November.

```
(a / accept-01
     :ARG0 (o / organization
          :ARG0 (n / name
               :op1 "IAEA"))
     :ARG1 (t2 / thing
          :ARG1-of (p / propose-01
               :ARG0 (c / country
                    :name (n2 / name
                         :op1 "North"
                         :op2 "Korea"))))
     :time (d / date-entity
               :month 11))
```

ISI Aligner

IAEA accepted North Korea 's proposal in November.

```
a / accept-01 :ARG0 o / organization :ARG0 n /
name :op1 "IAEA" :ARG1 t2 / thing :ARG1-of p /
propose-01 :ARG0 c / country :name n2 / name :op1
"North" :op2 "Korea" :time d / date-entity :month 11
```

Linearize the AMR using a
depth-first traversal

ISI Aligner

IAEA accepted North Korea proposal in November

```
accept IAEA propose North Korea :time 11 2001
```

English: remove stop words

AMR: remove special concepts, relations that don't usually align, quotes, and sense tags

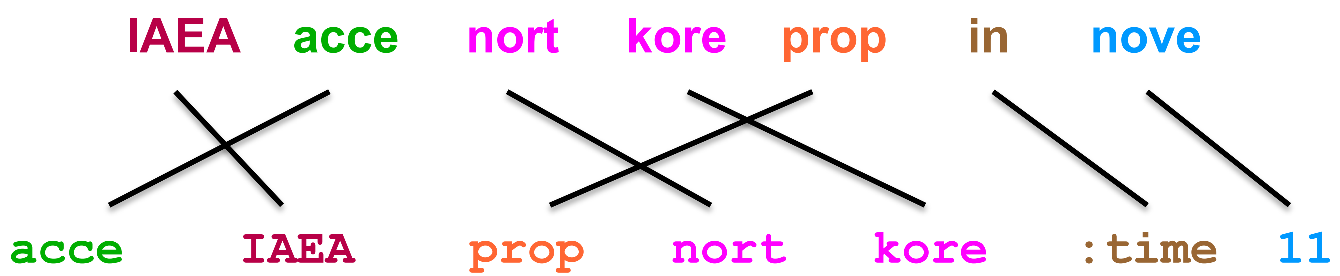
ISI Aligner

IAEA acce nort kore prop in nove

acce IAEA prop nort kore :time 11

Both: Lowercase and stem to the first four letters

ISI Aligner



Run IBM alignment models with a symmetrization constraint (write down), and project to AMR graph

Alignment: Summary

	JAMR aligner	ISI aligner
Alignment type	Graph fragment to span of words	Single concept to single word
Aligns edges	No	Yes
Learned from data	No	Yes
Gold standard available	https://github.com/jflanigan/jamr	http://amr.isi.edu/research.html
F_1 score*	90%	83%

*not directly comparable, since different gold standard

In general, the desired type of alignment will depend on the application

Parsing

- Evaluation
- Alignment
- **Parsing**
  - Graph-based parsing
    - Structured prediction
    - Concept identification
    - Relation identification
      - Maximum spanning connected graph algorithm (MSCG)
      - Graph determinism constraints using Lagrangian relaxation
    - Experiments
  - Transition-based parsing
- Graph Grammars and Automata
- Applications

JAMR Overview (Flanigan et al, 2014)

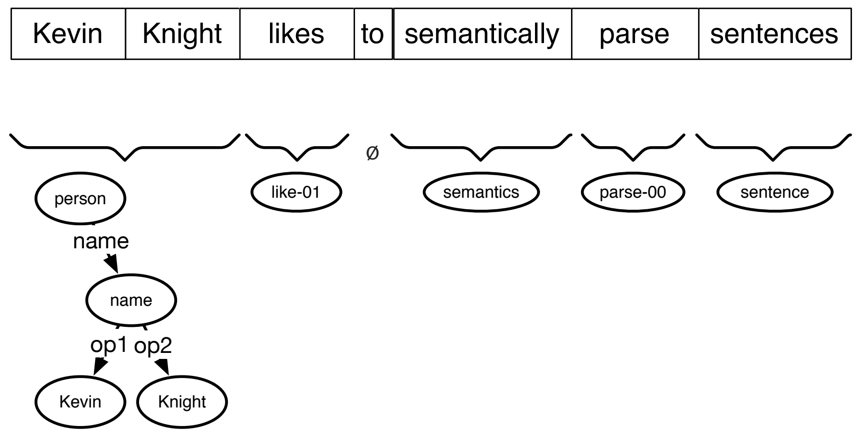
Input

Kevin	Knight	likes	to	semantically	parse	sentences

JAMR Overview

Input

Concept ID

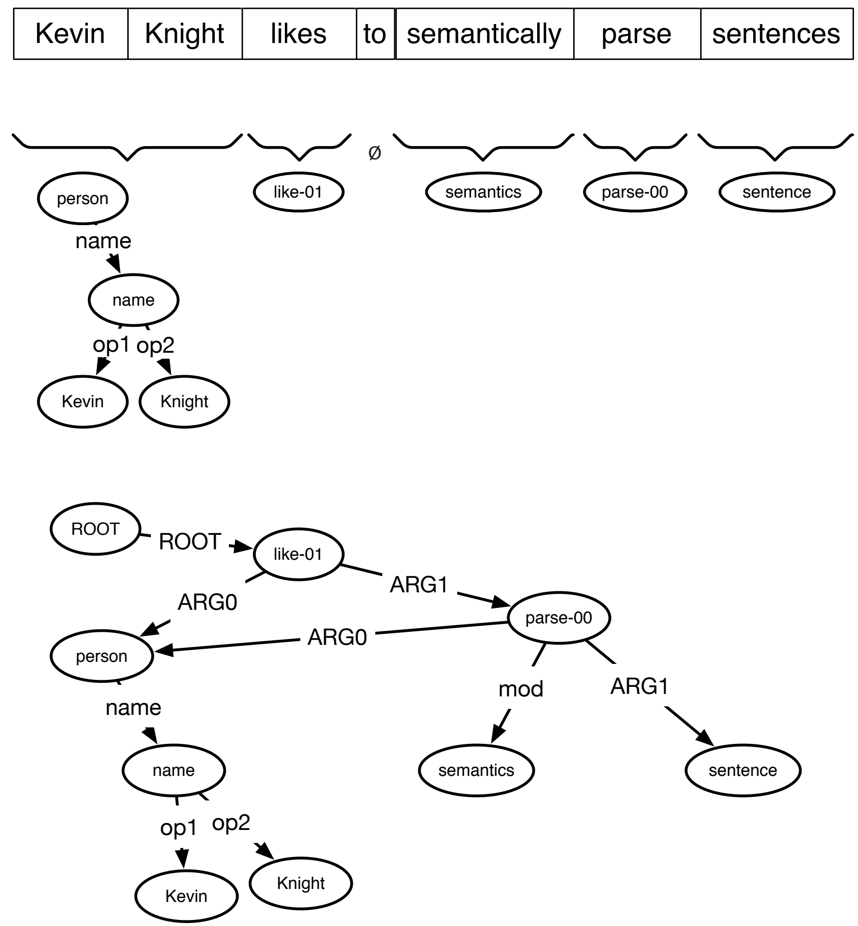


JAMR Overview

Input

Concept ID

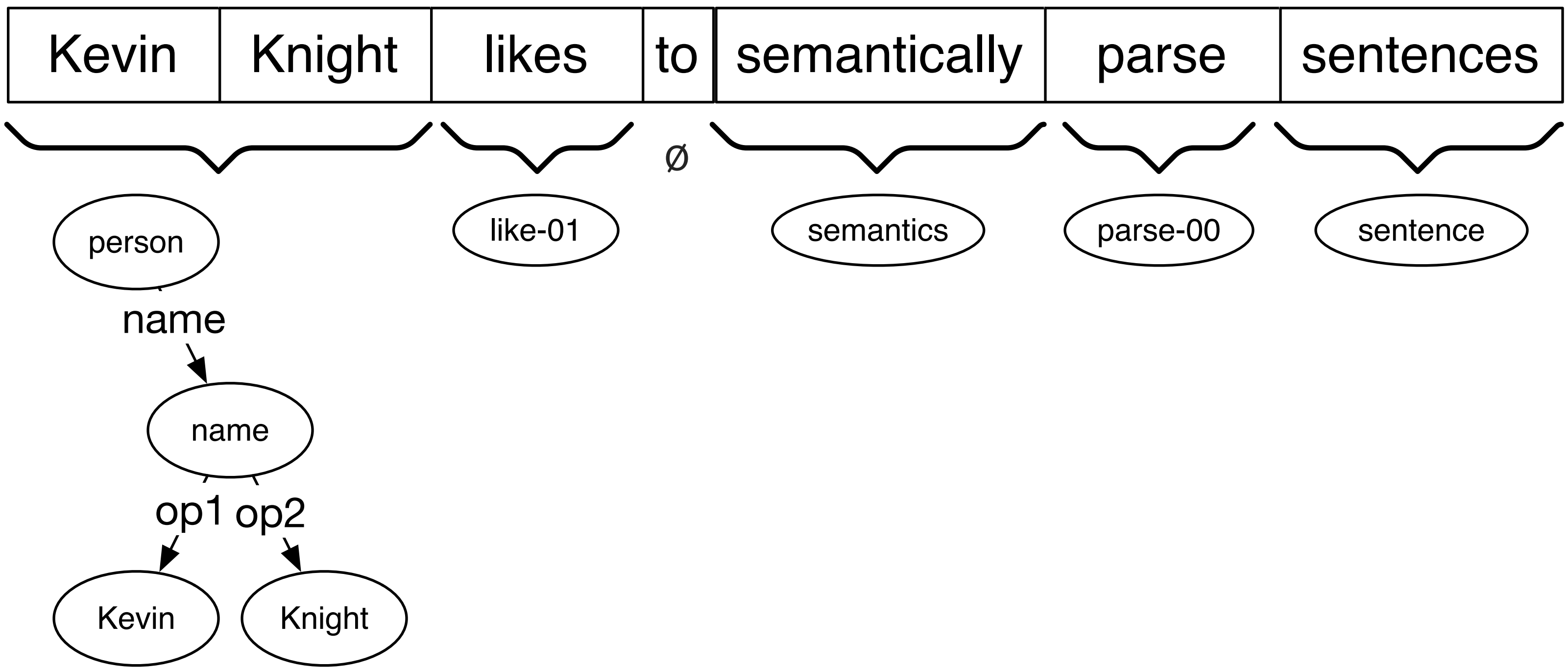
Relation ID



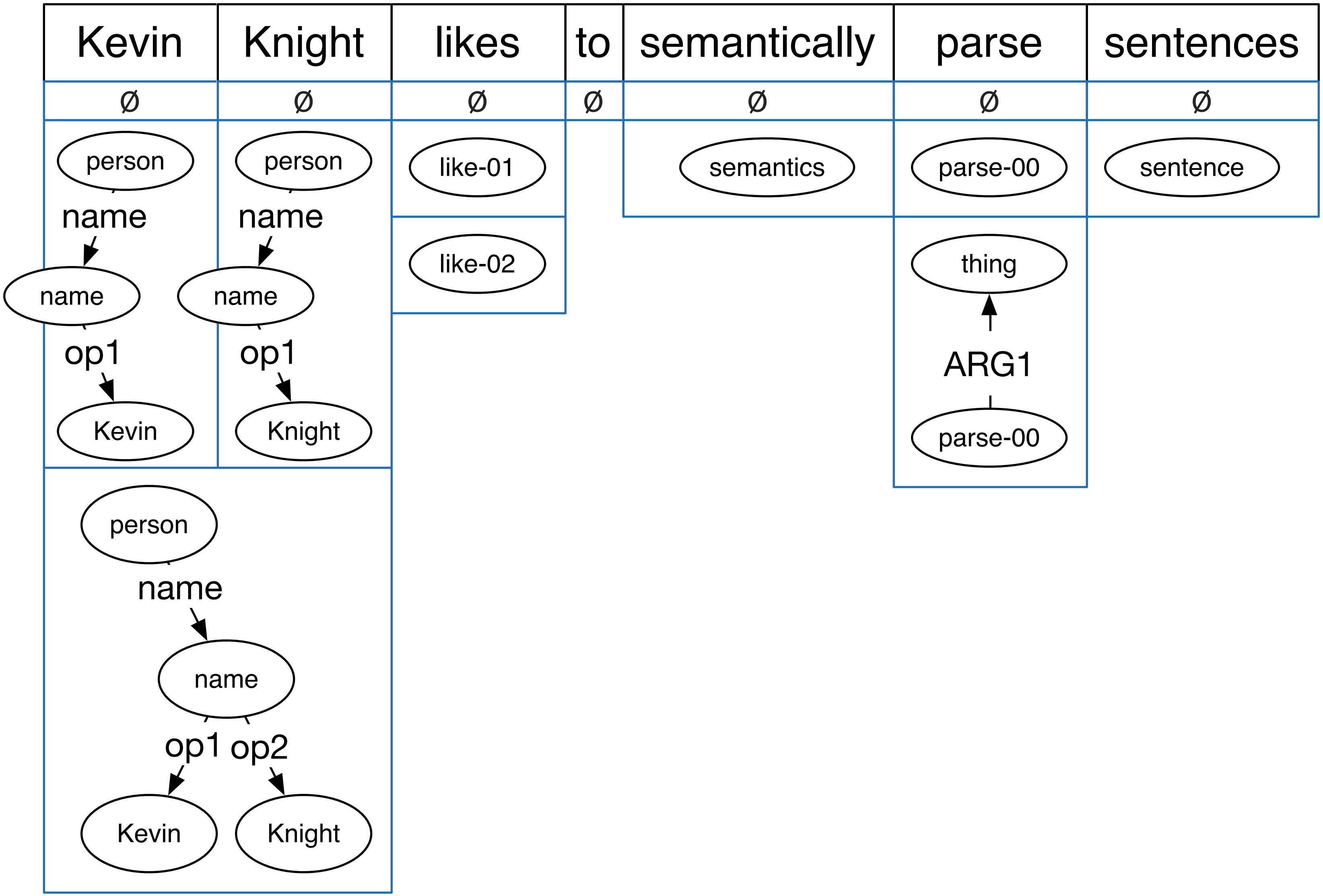
Concept Identification

Kevin	Knight	likes	to	semantically	parse	sentences

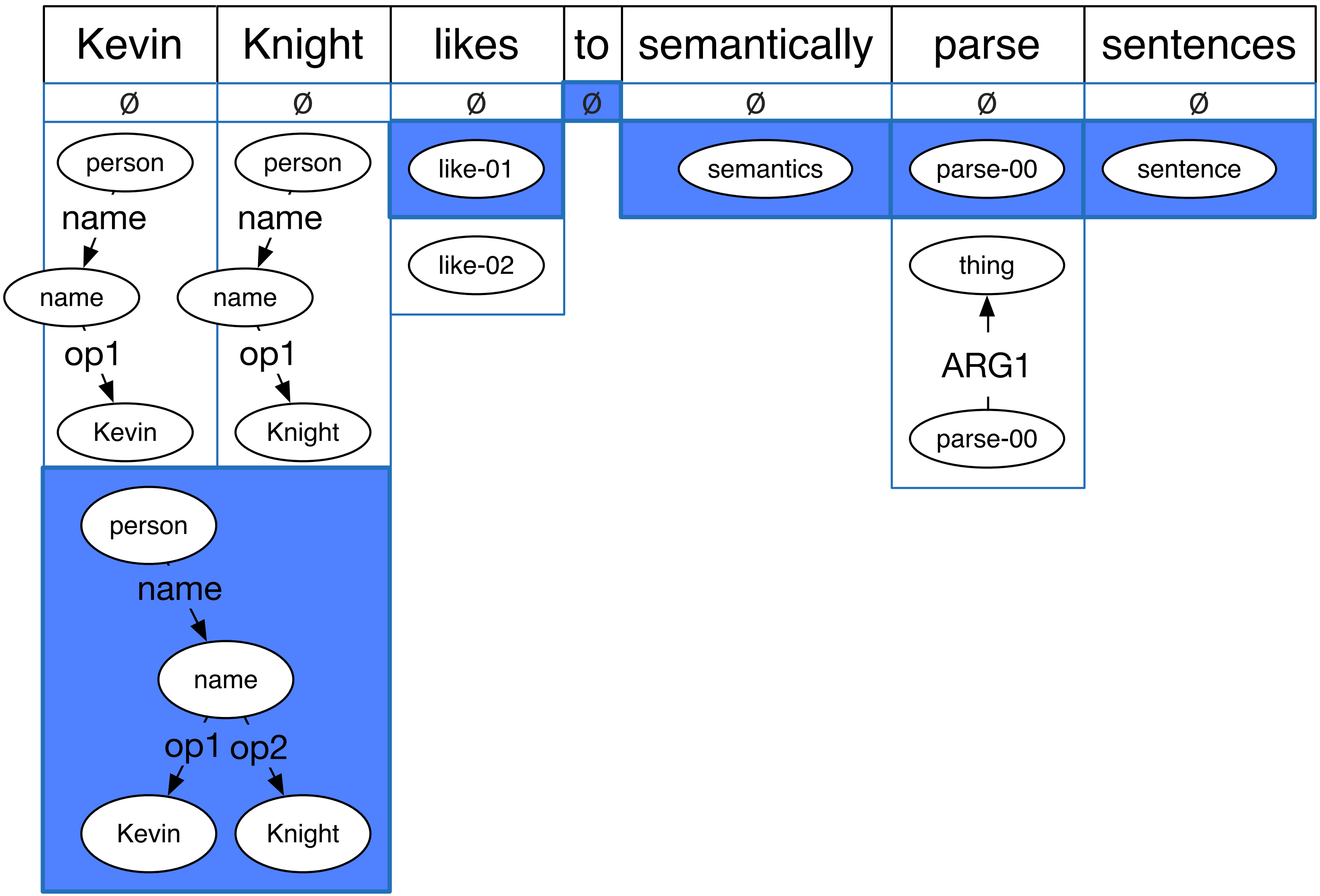
Concept Identification



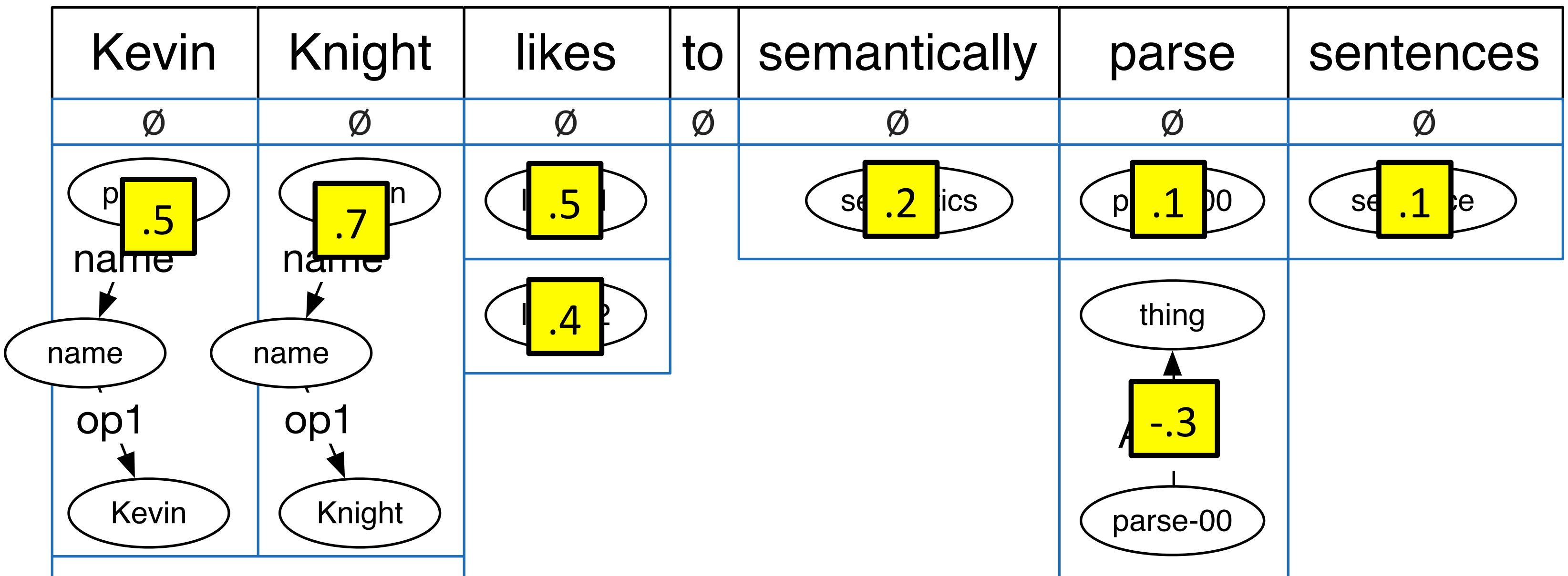
Concept Identification

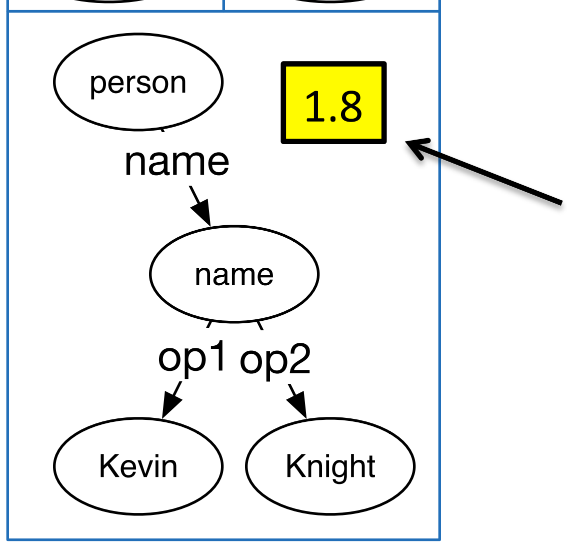


Concept Identification



Concept Identification

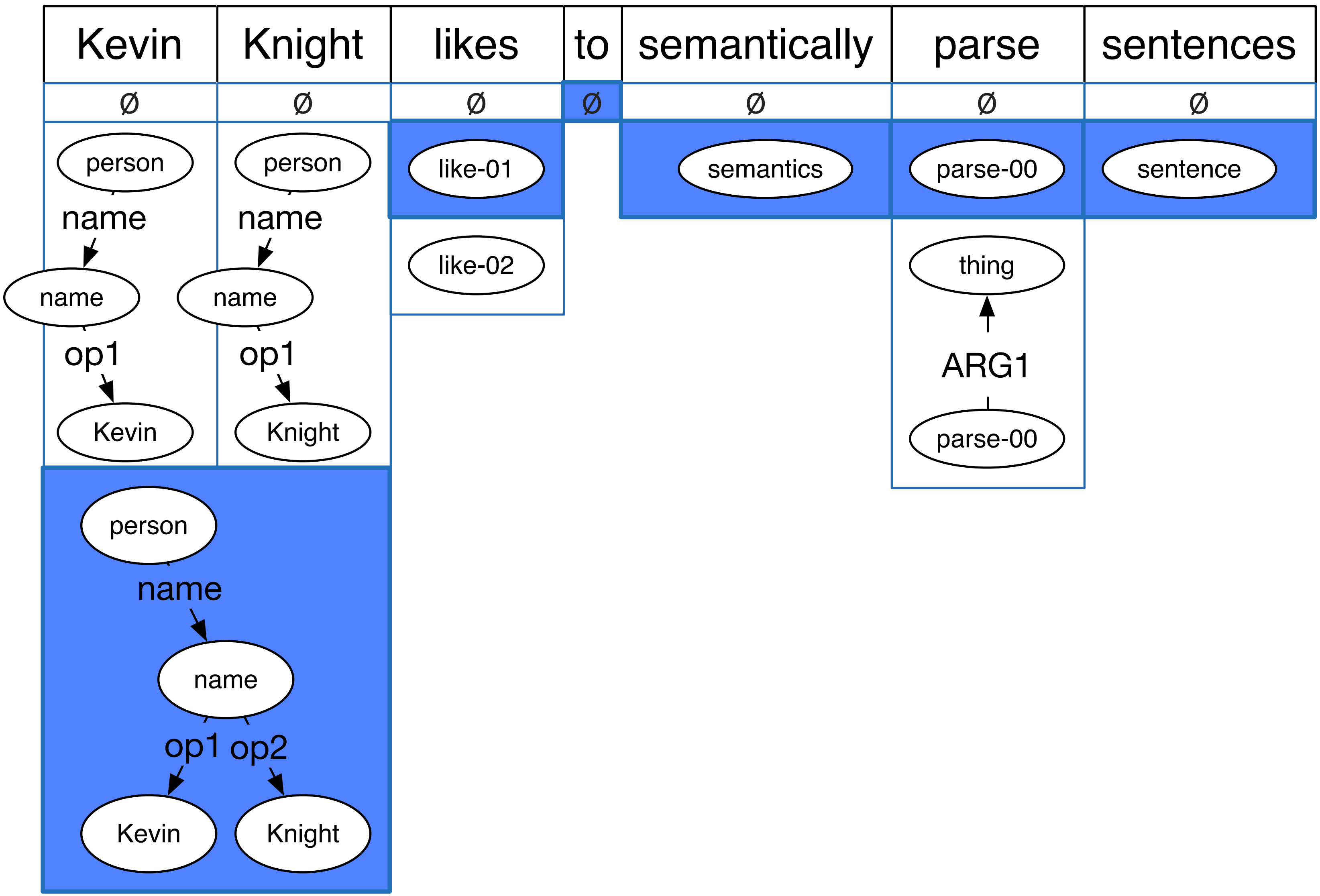




Local score = how much the model wants to include this fragment

= *model_weights · local_features*

Concept Identification



Training

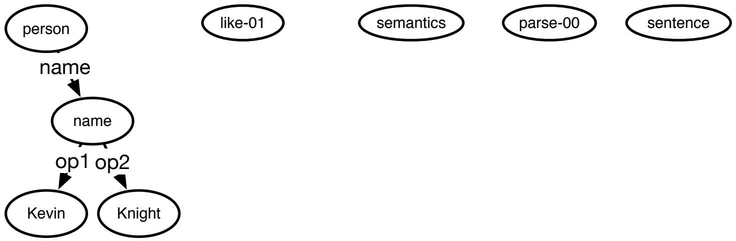
- AdaGrad structured perceptron

Formula for updating the weights

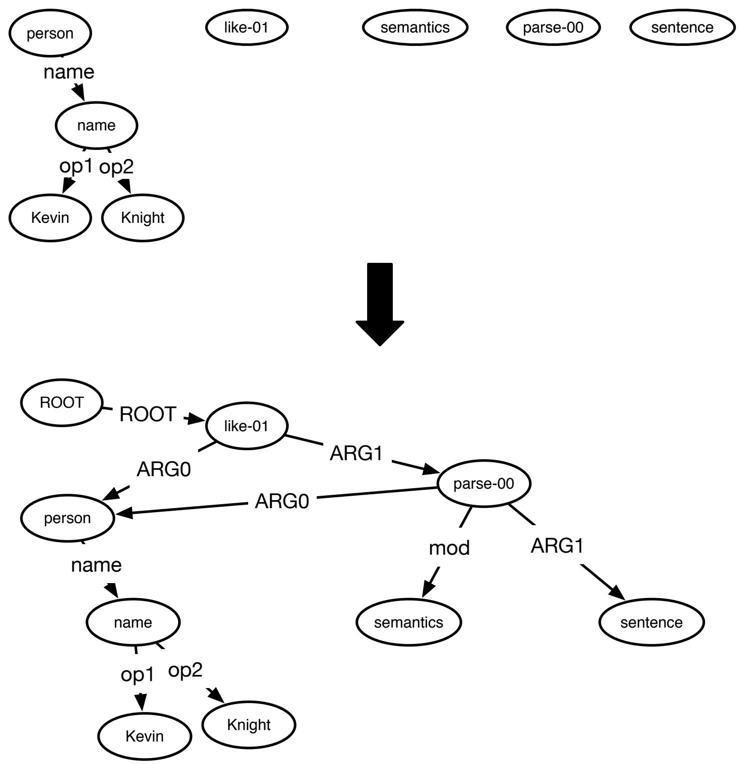
Relation Identification

- Evaluation
- Alignment
- Parsing
  - Graph-based parsing
    - Concept identification
    - **Relation identification**
      - Maximum spanning connected graph algorithm (MSCG)
      - Graph determinism constraints using Lagrangian relaxation
    - Experiments
  - Transition-based parsing
- Graph Grammars and Automata
- Applications

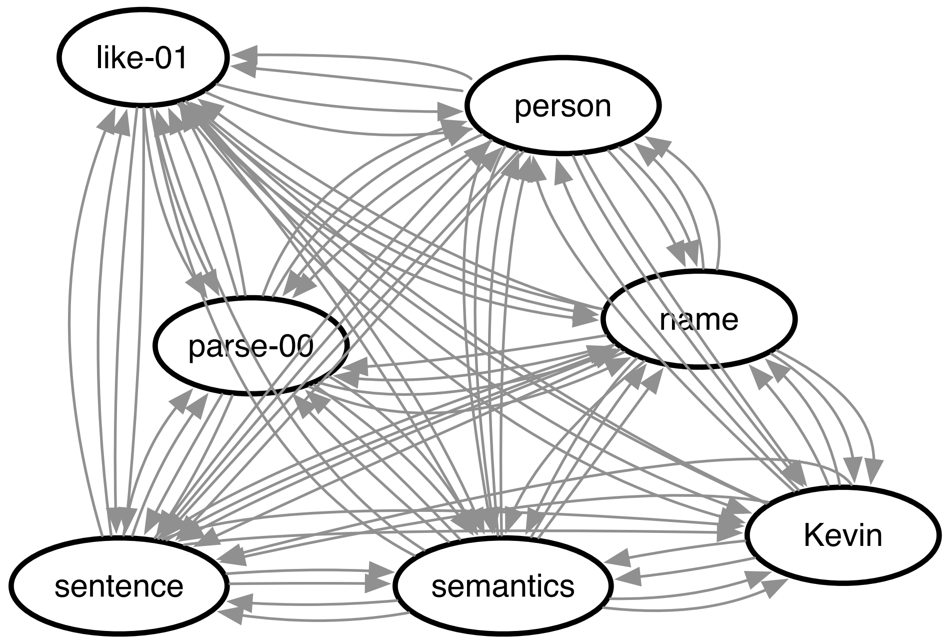
Relation Identification



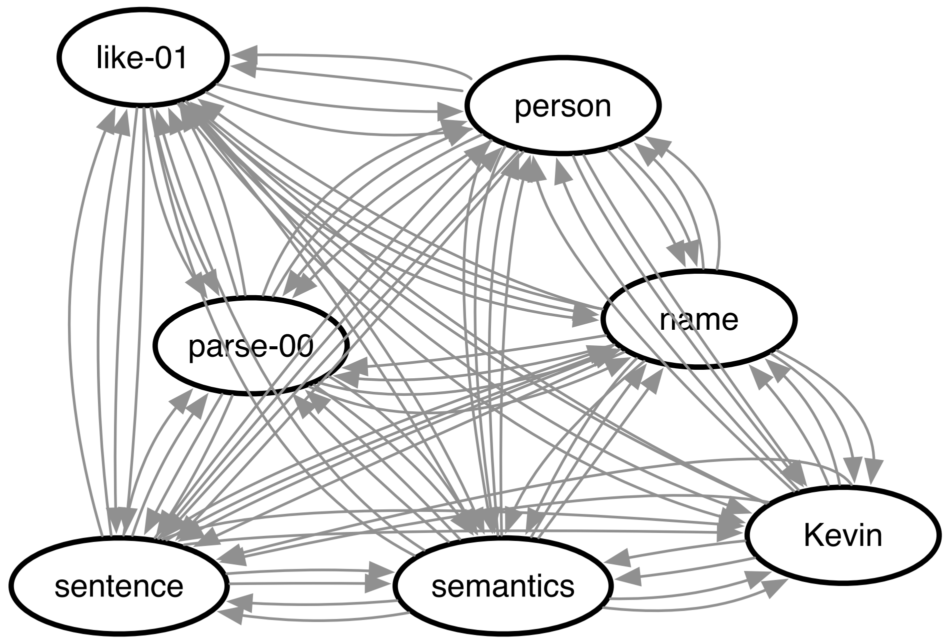
Relation Identification



Dense Graph

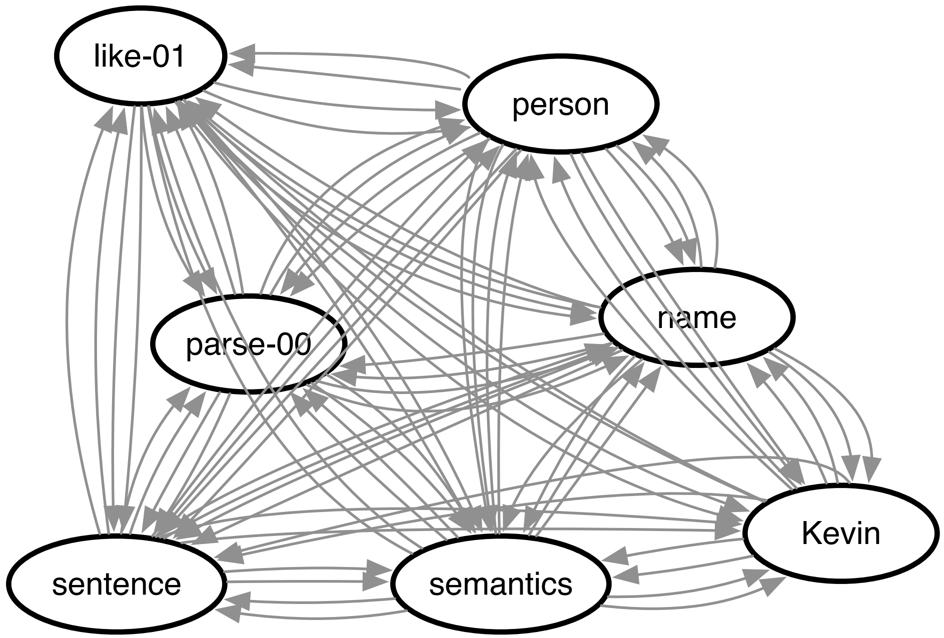


Dense Graph



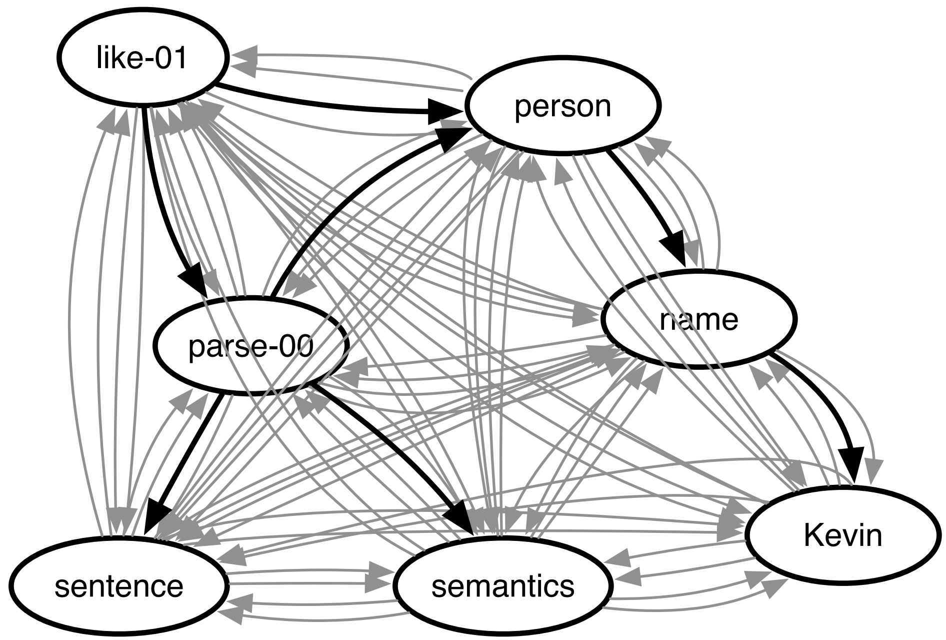
- All possible edges between all nodes
- Edges w/ weights

Dense Graph



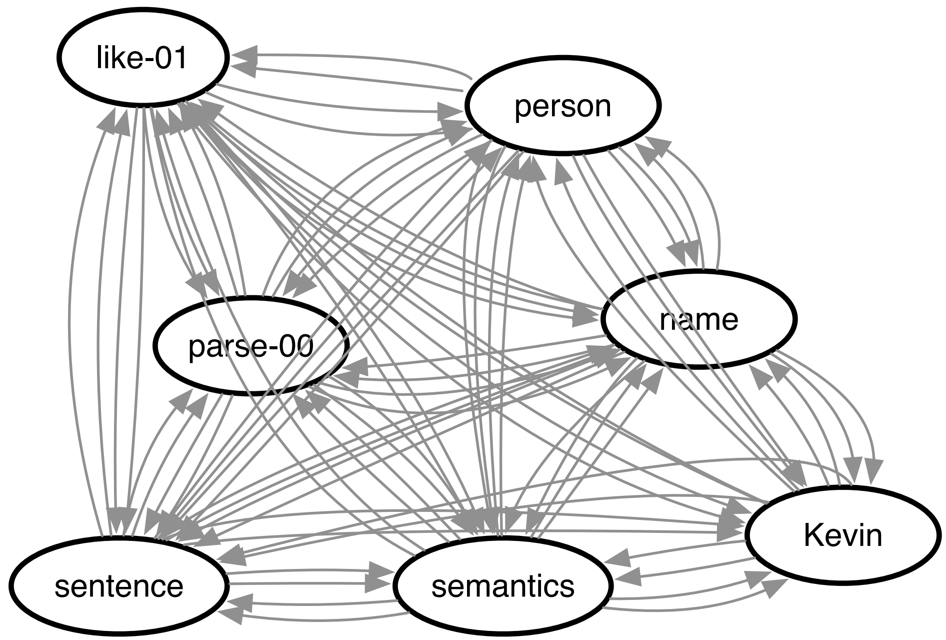
Edge weight = how much the model wants to include that edge in the output graph

Dense Graph



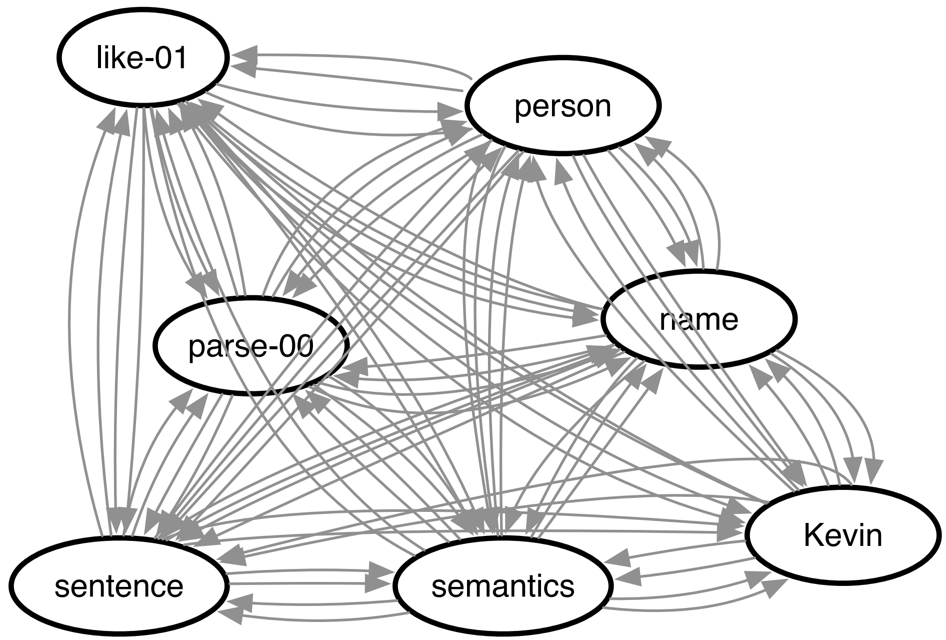
Output graph = max subgraph with constraints on well-formedness

Dense Graph



$\mathbf{z}$ binary vector, indicates which edges are selected

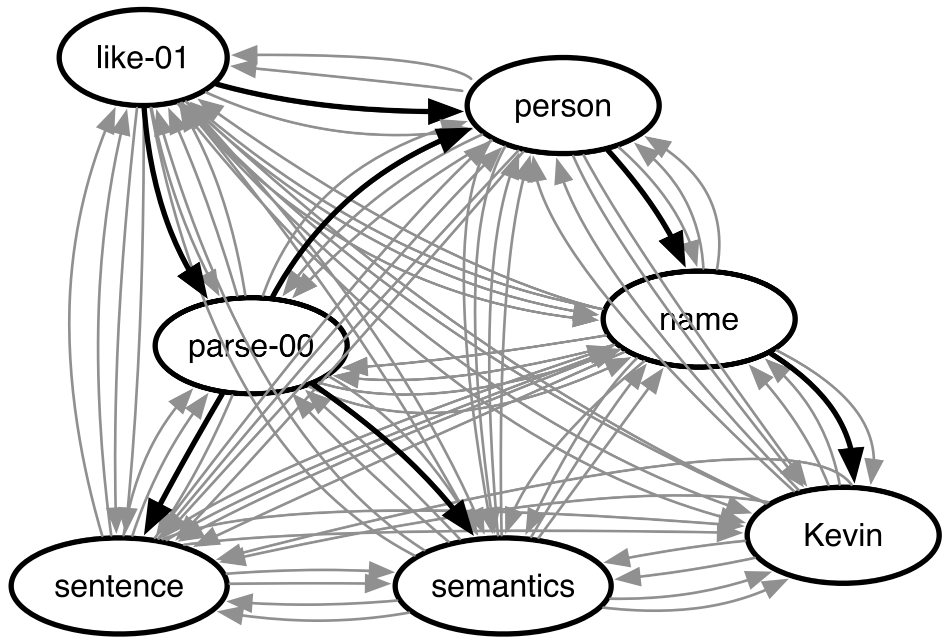
Dense Graph



Sum of edge weights = $\phi^T \mathbf{z}$

Vector which contains the edge weights

Max Subgraph



Can write relation ID as optimization problem $\max_{\mathbf{z} \in \mathcal{Z}} \boldsymbol{\phi}^T \mathbf{z}$

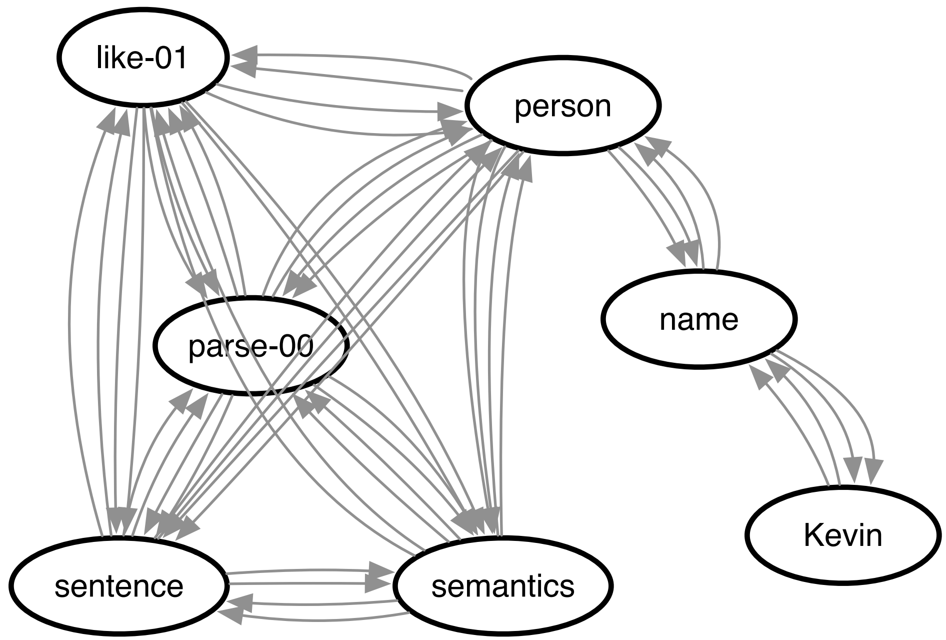
Set of graphs satisfying the constraints

Output Graph Properties (Constraints)

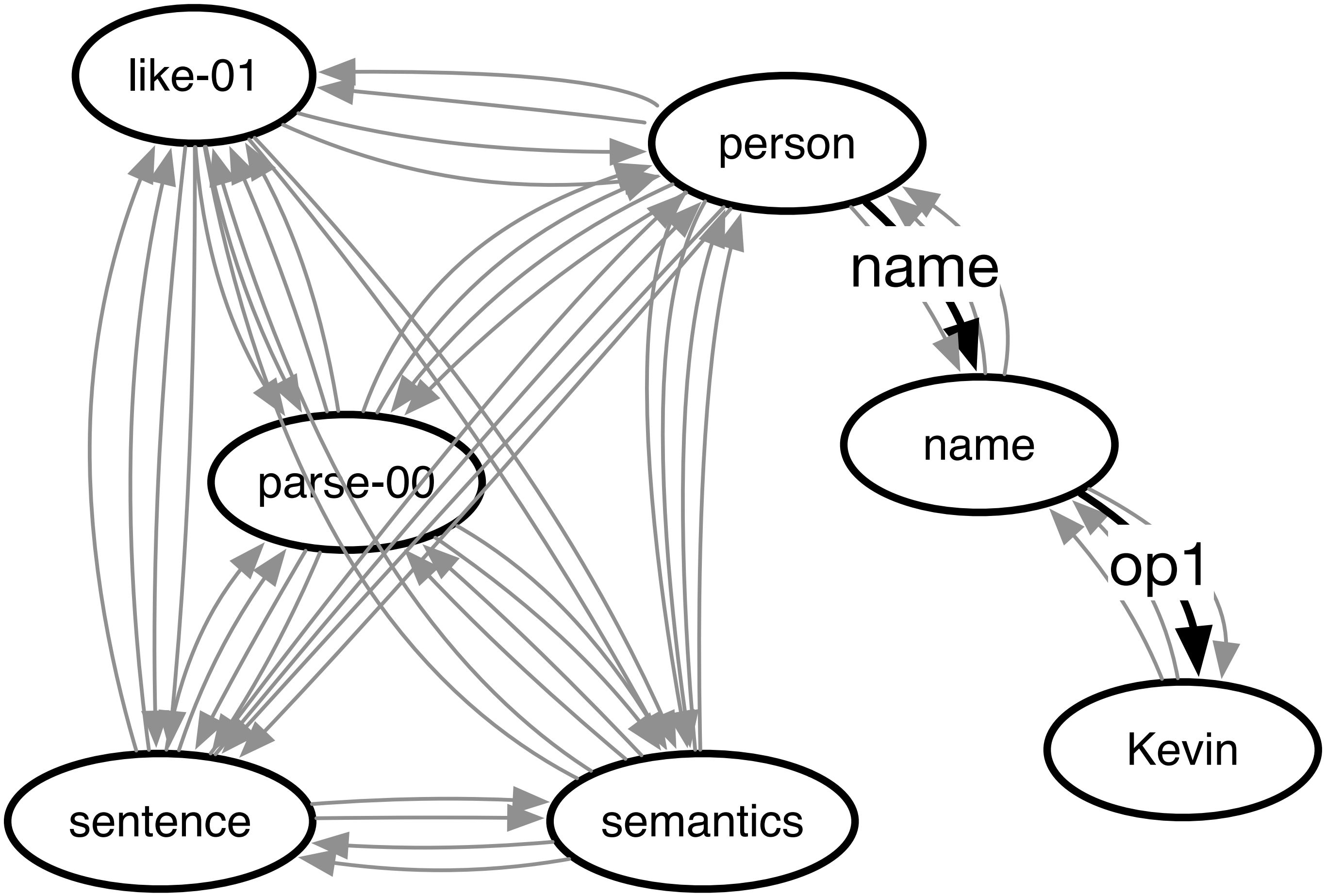
- Preserving
- Simple
- Spanning (all nodes)
- Connected
- Deterministic

(picture of graph, no text in node)

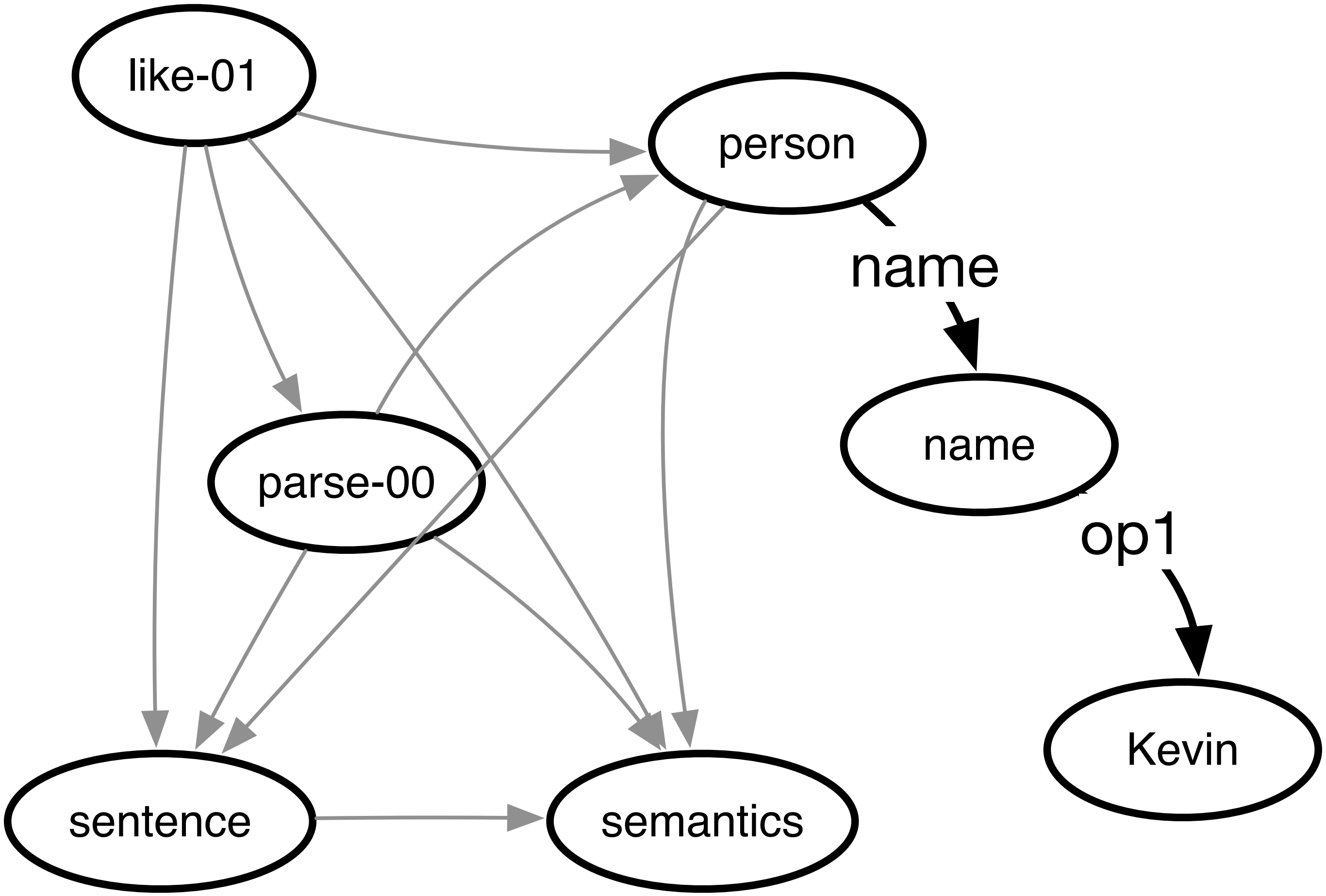
Reduced graph for clarity



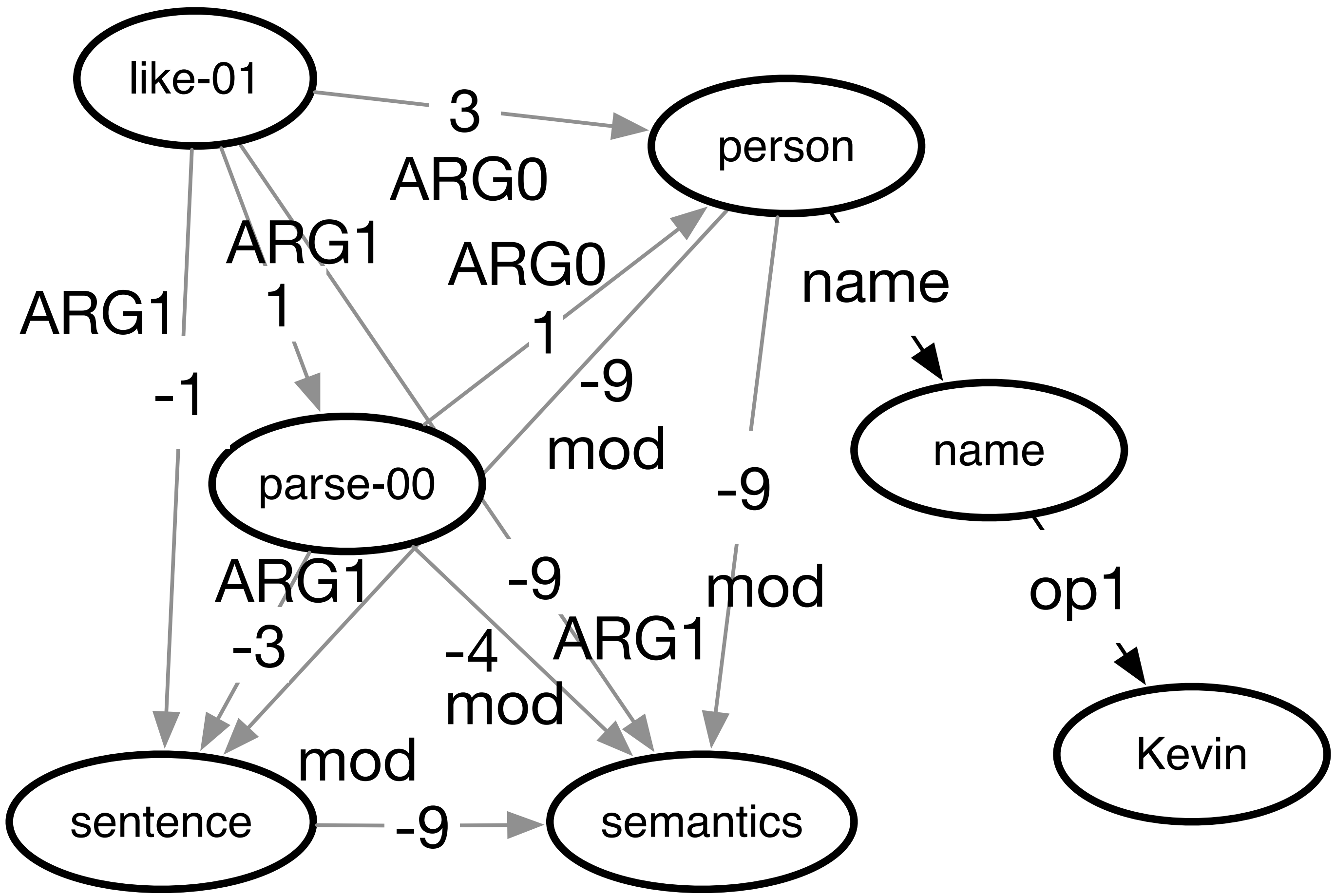
Constraint: Preserving



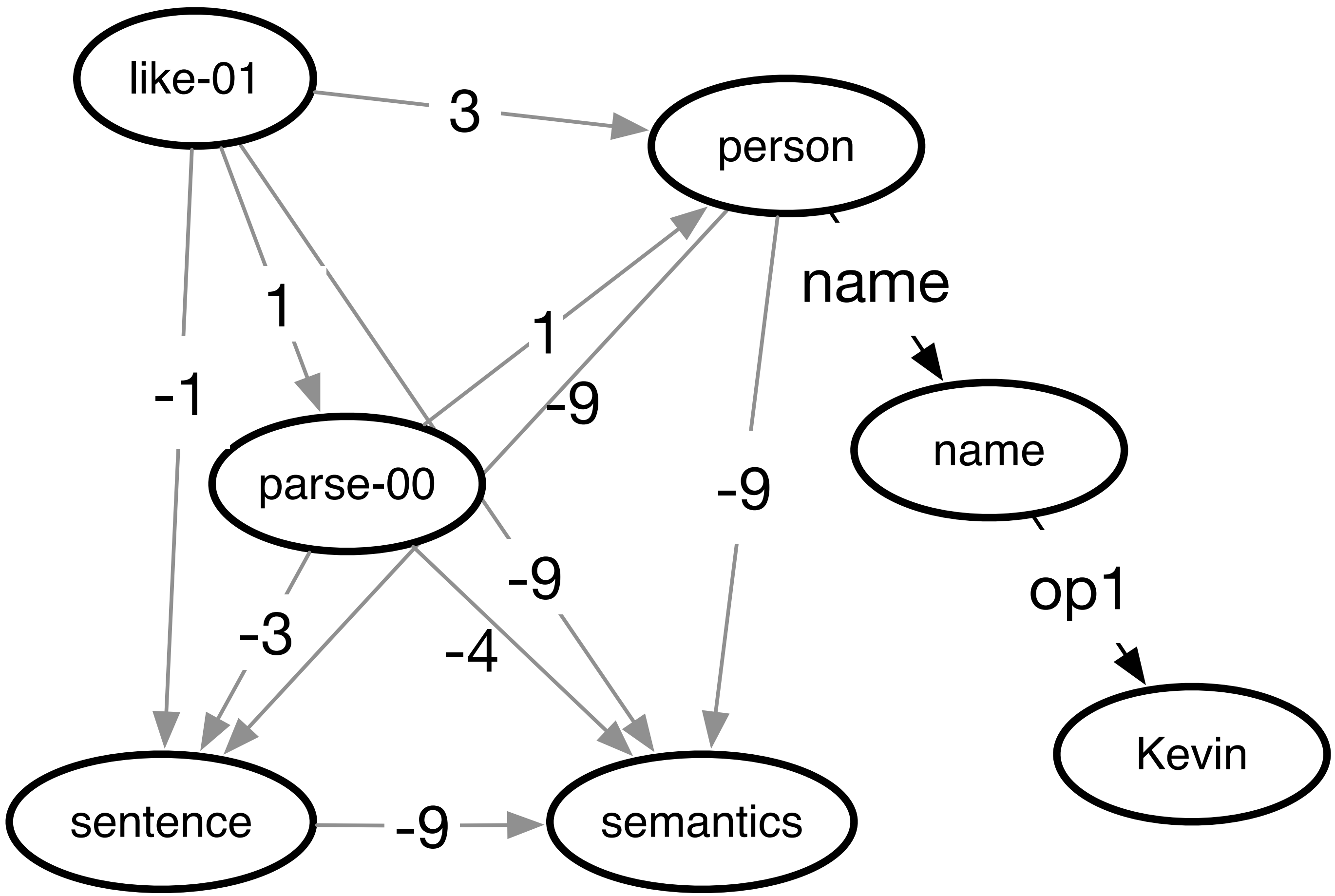
Constraint: Simple



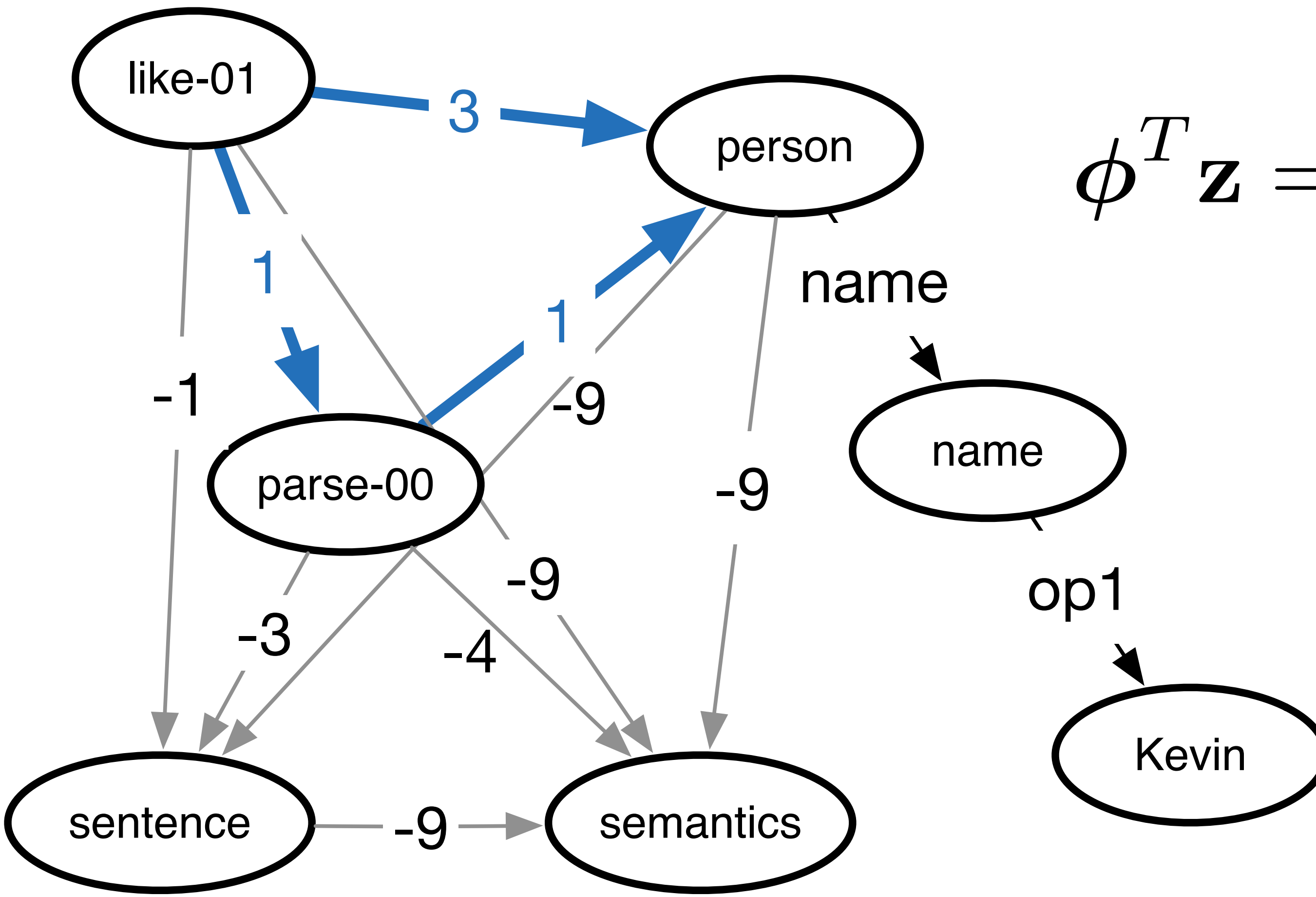
With Weights and Labels Shown



Maximum Weighted Subgraph

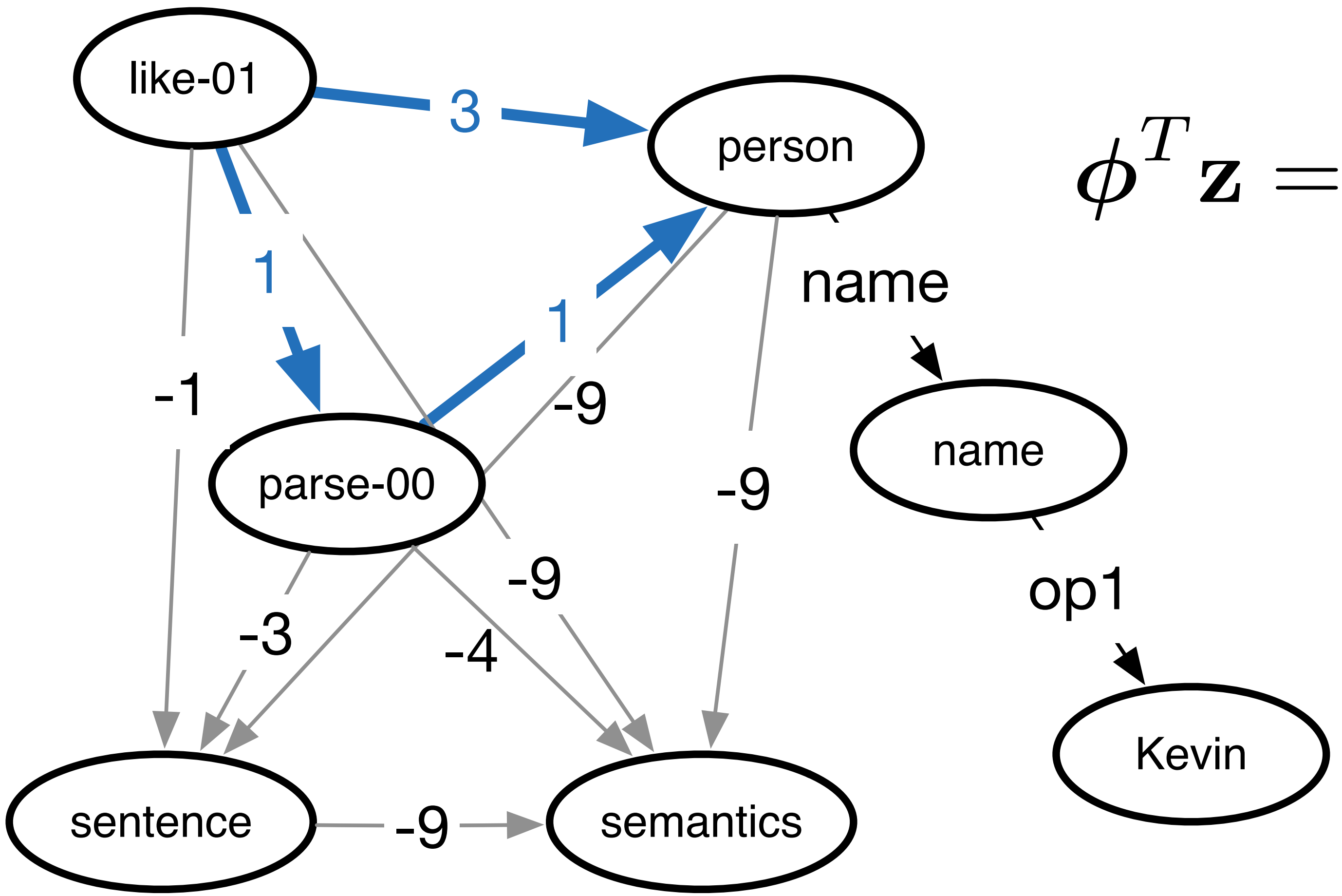


Maximum Weighted Subgraph



$$\phi^T \mathbf{z} = 5$$

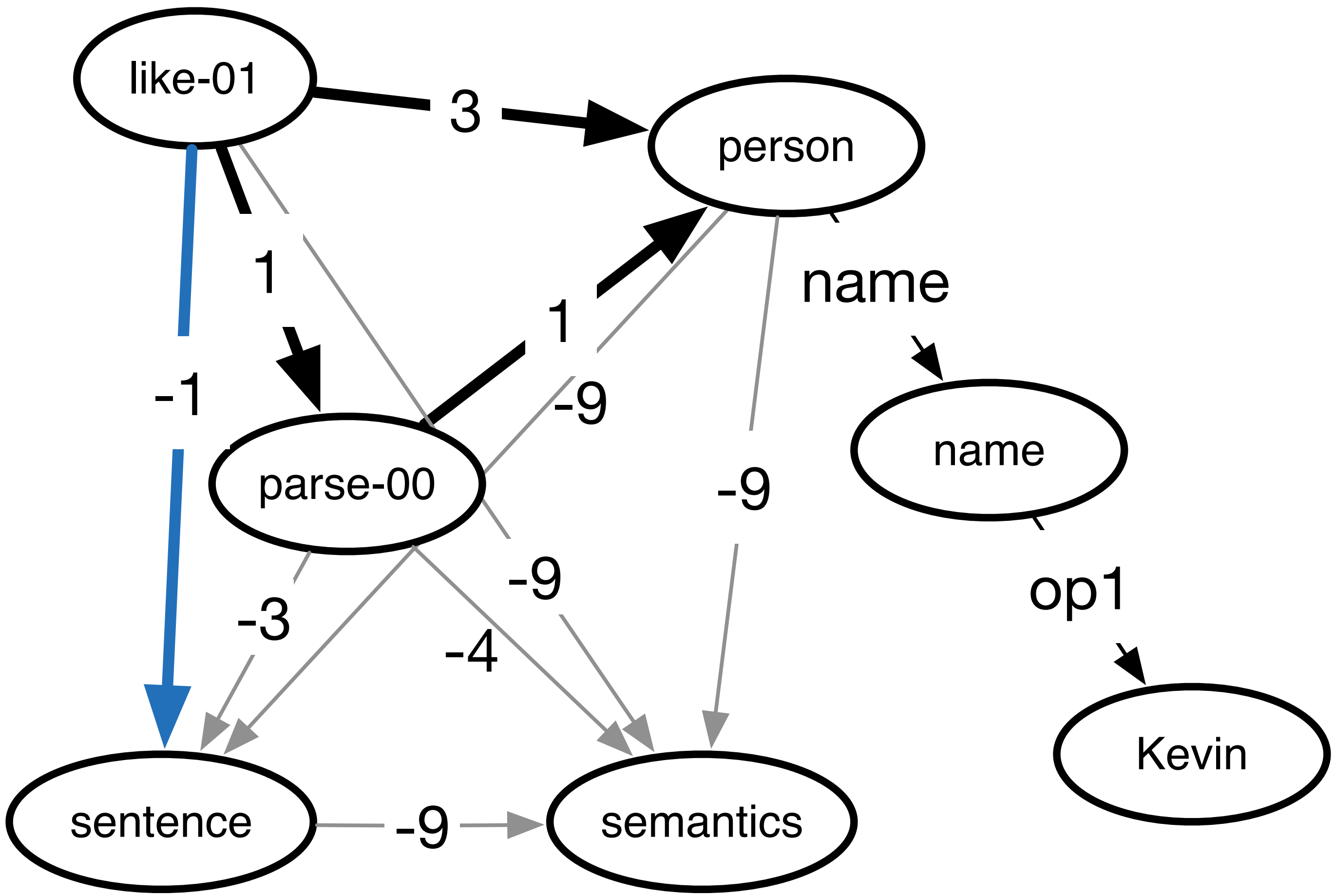
Maximum Weighted Subgraph



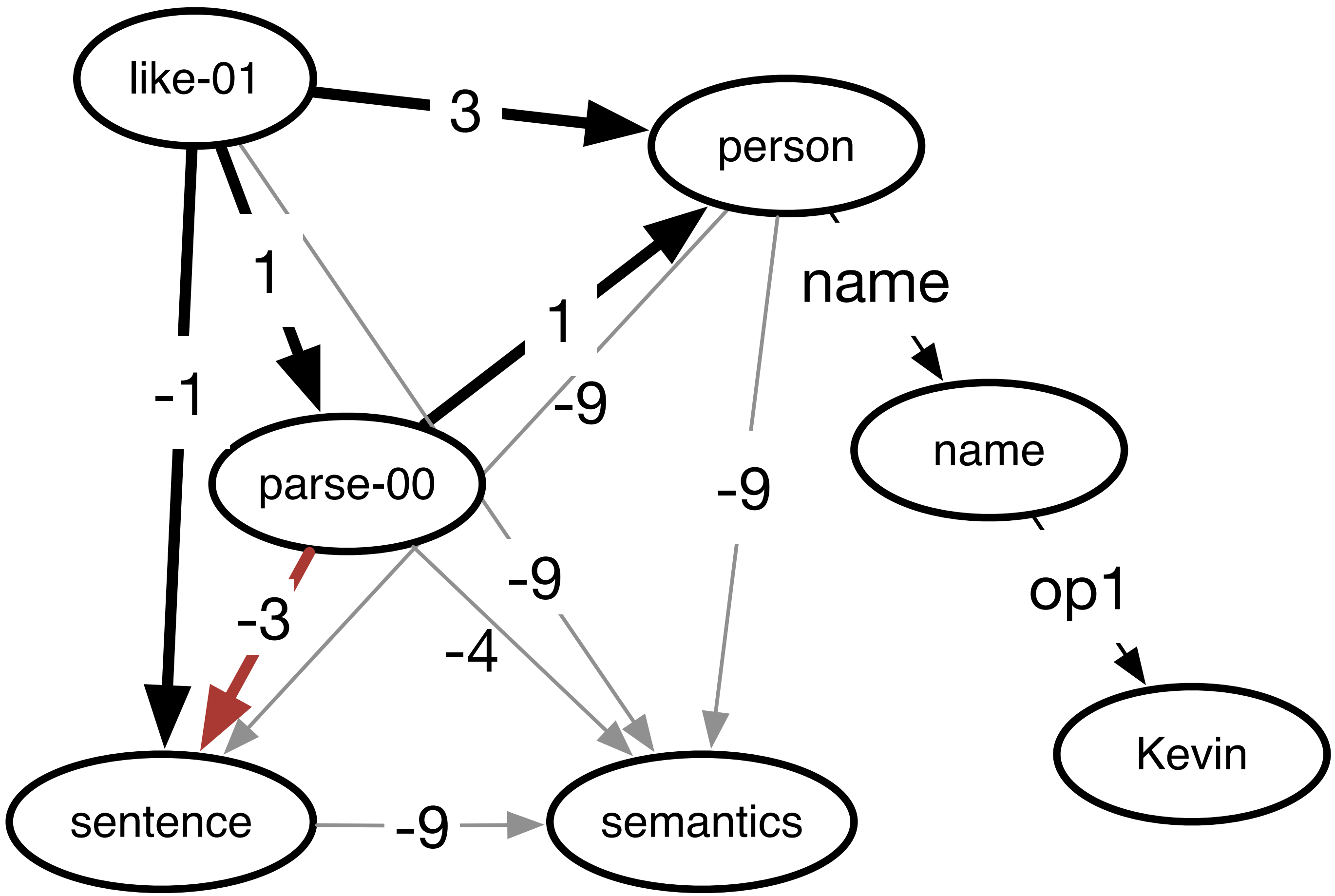
$$\phi^T \mathbf{z} = 5$$

Constraint: Graph must be connected

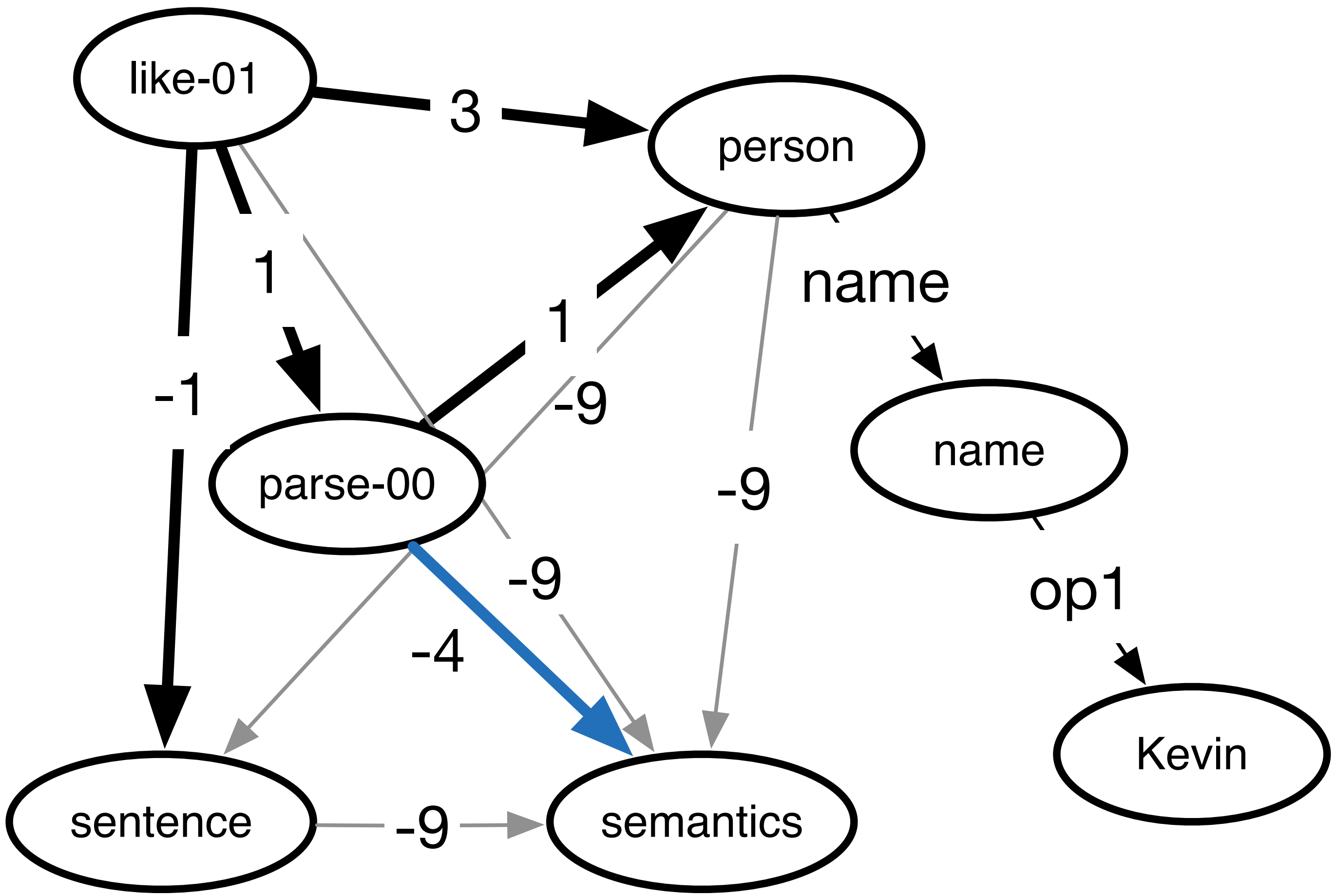
Maximum Spanning, Connected Subgraph (MSCG)



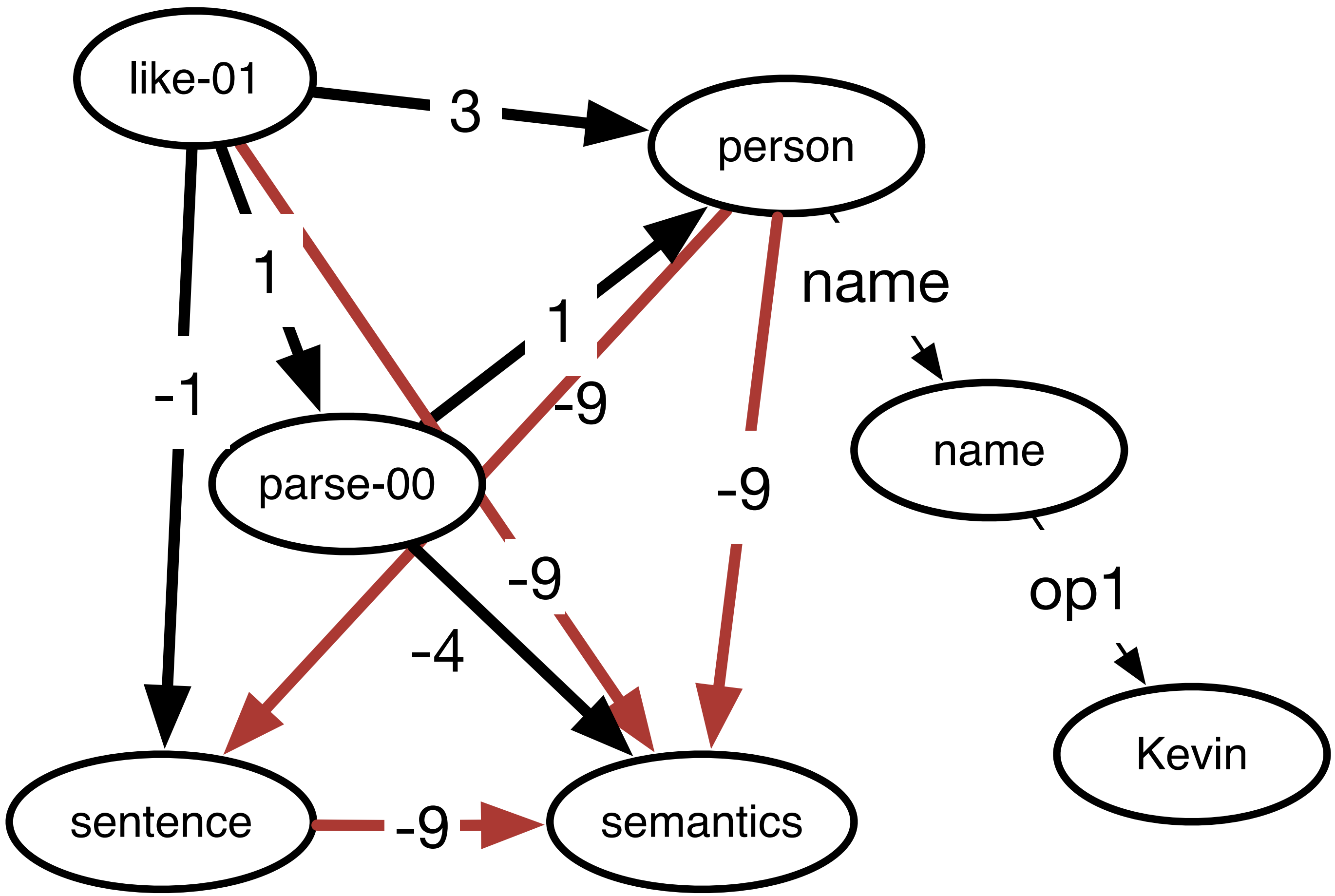
Maximum Spanning, Connected Subgraph (MSCG)



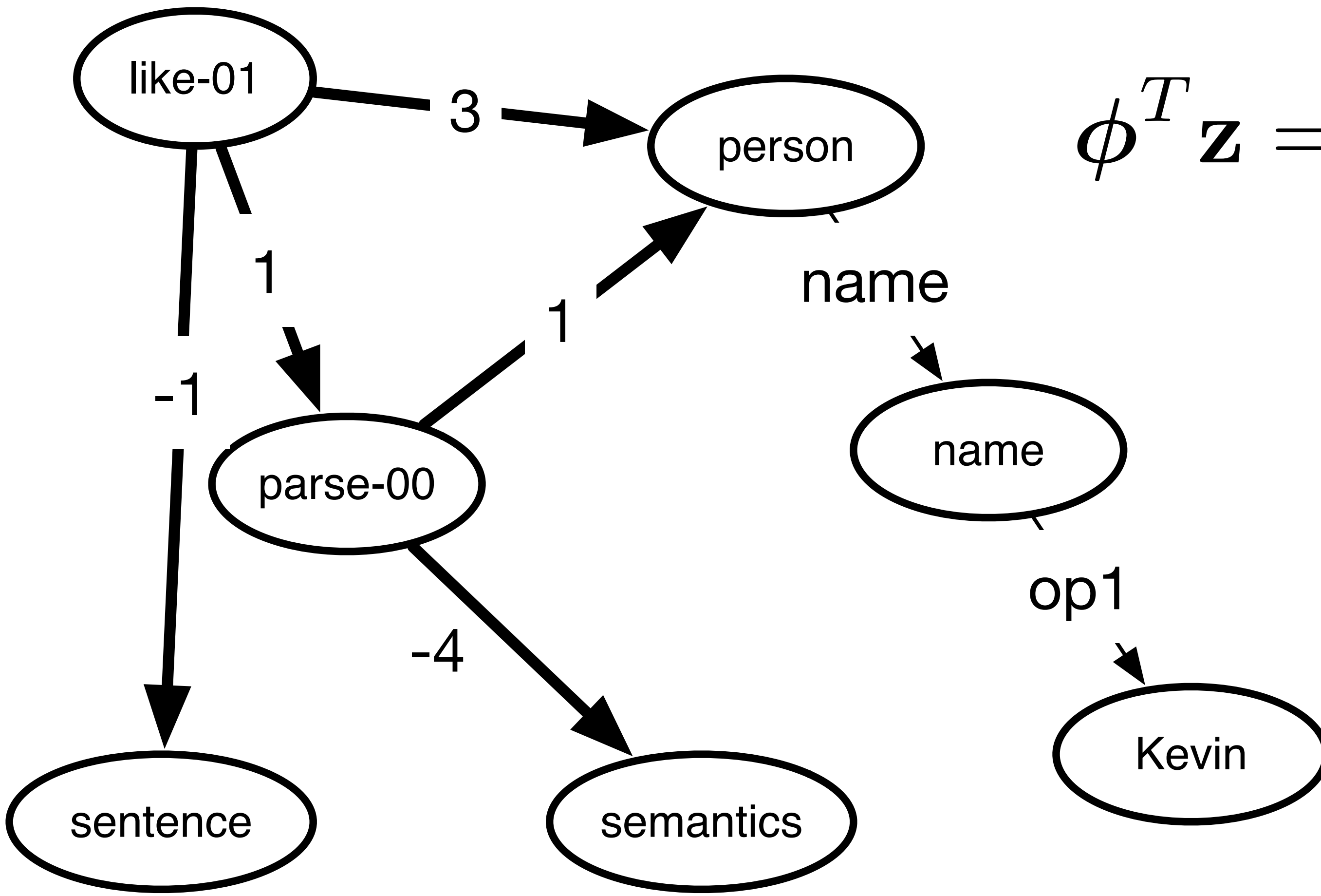
Maximum Spanning, Connected Subgraph (MSCG)



Maximum Spanning, Connected Subgraph (MSCG)

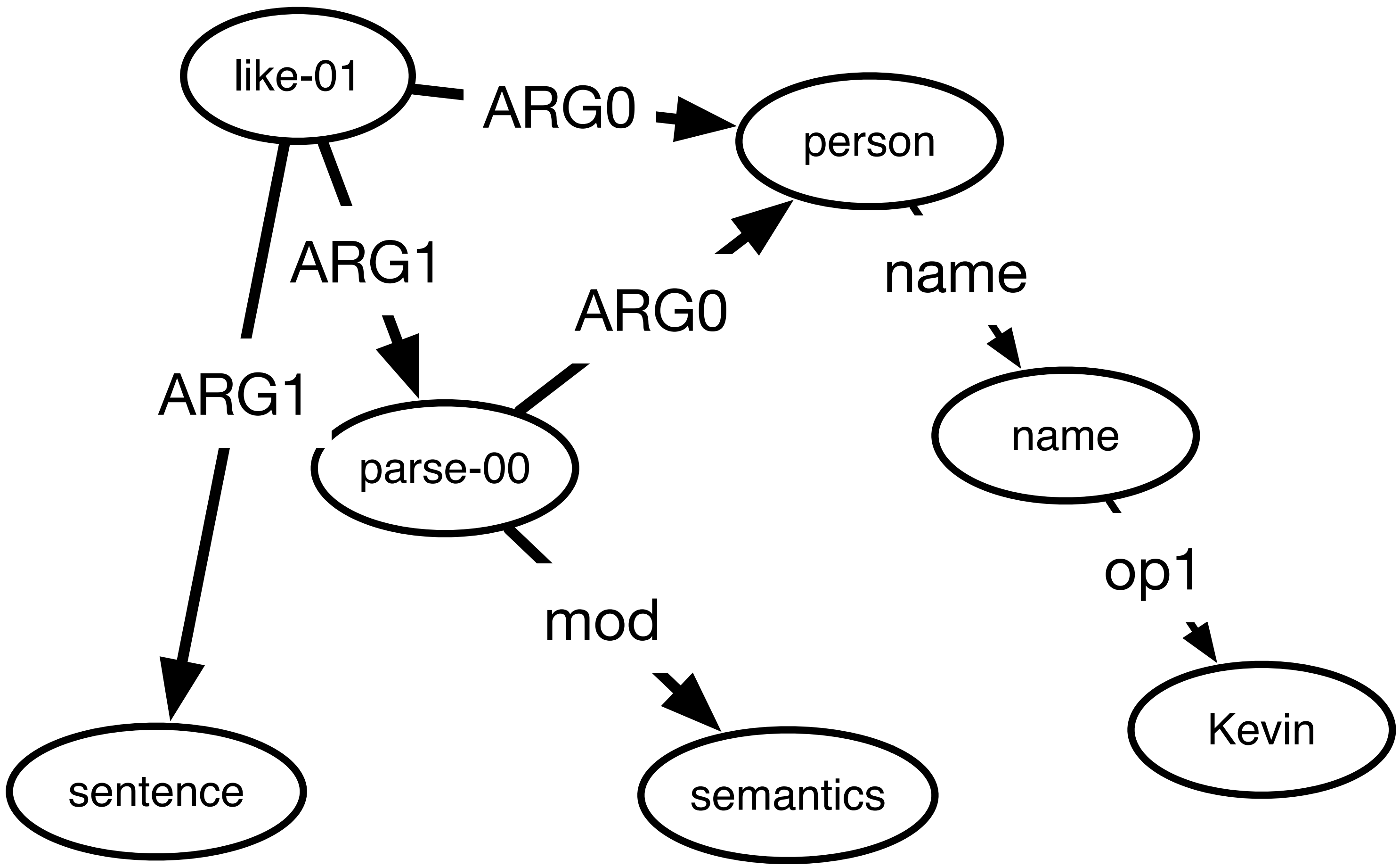


Maximum Spanning, Connected Subgraph (MSCG)



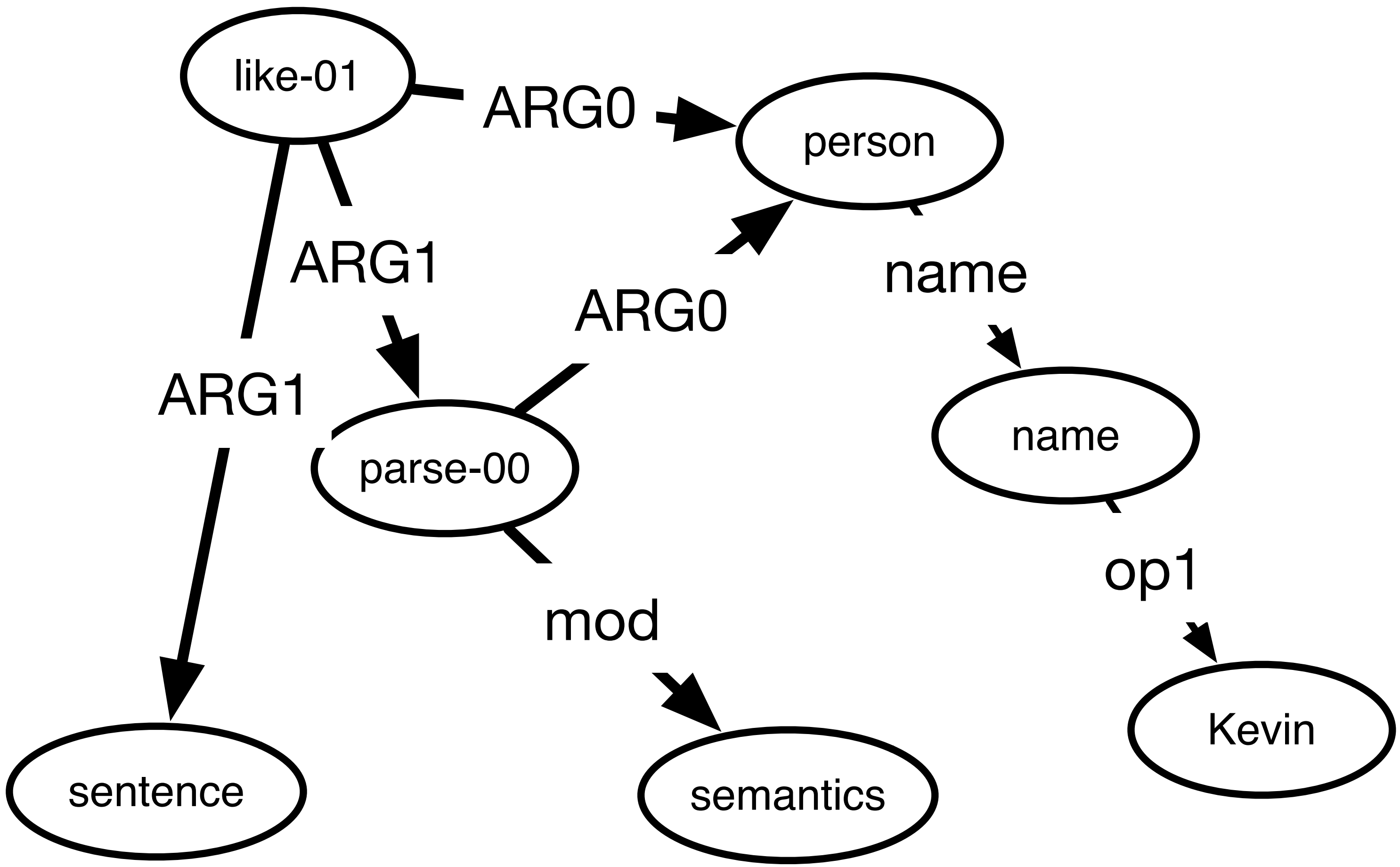
$$\phi^T \mathbf{z} = 0$$

Maximum Spanning, Connected Subgraph (MSCG)



Constraint: Graph must be deterministic

Determinism Constraints



Determinism Constraints

$$\mathbf{z}_{1 \xrightarrow{\text{ARG1}} 2} + \mathbf{z}_{1 \xrightarrow{\text{ARG1}} 3} + \ldots \leq 1$$

$$\mathbf{z}_{2 \xrightarrow{\text{ARG1}} 1} + \mathbf{z}_{2 \xrightarrow{\text{ARG1}} 3} + \ldots \leq 1$$

$$\vdots$$

Determinism Constraints

$$\mathbf{z}_{1 \xrightarrow{\text{ARG1}} 2} + \mathbf{z}_{1 \xrightarrow{\text{ARG1}} 3} + \ldots \leq 1$$

$$\mathbf{z}_{2 \xrightarrow{\text{ARG1}} 1} + \mathbf{z}_{2 \xrightarrow{\text{ARG1}} 3} + \ldots \leq 1$$

$$\vdots$$

$$A\mathbf{z} \leq b$$

New Objective

$$\max_{\mathbf{z} \in \mathcal{Z}} \boldsymbol{\phi}^T \mathbf{z}$$

s.t. $\mathbf{z}$ satisfies $A\mathbf{z} \leq b$

Lagrangian Relaxation

$$\max_{\mathbf{z} \in \mathcal{Z}} \boldsymbol{\phi}^T \mathbf{z} + \boldsymbol{\lambda}^T (\mathbf{b} - A\mathbf{z})$$

Lagrangian Relaxation

$$\max_{\mathbf{z} \in \mathcal{Z}} \boldsymbol{\phi}^T \mathbf{z} + \boldsymbol{\lambda}^T (\mathbf{b} - A\mathbf{z})$$

$$= \max_{\mathbf{z} \in \mathcal{Z}} (\boldsymbol{\phi}^T - \boldsymbol{\lambda}^T A)\mathbf{z} + const$$

Lagrangian Relaxation

$$\max_{\mathbf{z}\in\mathcal{Z}} \boldsymbol{\phi}^T\mathbf{z} + \boldsymbol{\lambda}^T(\mathbf{b} - A\mathbf{z})$$
$$= \max_{\mathbf{z}\in\mathcal{Z}} (\boldsymbol{\phi}^T - \boldsymbol{\lambda}^T A)\mathbf{z} + const$$

Minimize over $\boldsymbol{\lambda}$ with sub-grad descent

After many iterations

The image cannot be displayed. Your computer may not have enough memory to open the image, or the image may have been corrupted. Restart your computer, and then open the file again. If the red x still appears, you may have to delete the image and then insert it again.

Summary: Output Graph Properties

- Maximum weight
- Preserving
- Simple
- Spanning (all nodes)
- Connected
- Deterministic

Features & Training

- (list of features)
- Trained using AdaGrad structured perceptron

Experiments

- Evaluation
- Alignment
- Parsing
  - Graph-based parsing
    - Concept identification
    - Relation identification
      - Maximum spanning connected graph algorithm (MSCG)
      - Graph determinism constraints using Lagrangian relaxation
    - **Experiments**
  - Transition-based parsing
- Graph Formalisms
- Applications

Experimental Setup

- Data: LDC2013E117
  - 4,000 training instances
  - 2,000 test
  - 2,000 dev
- Split: Flanigan et al (2014)

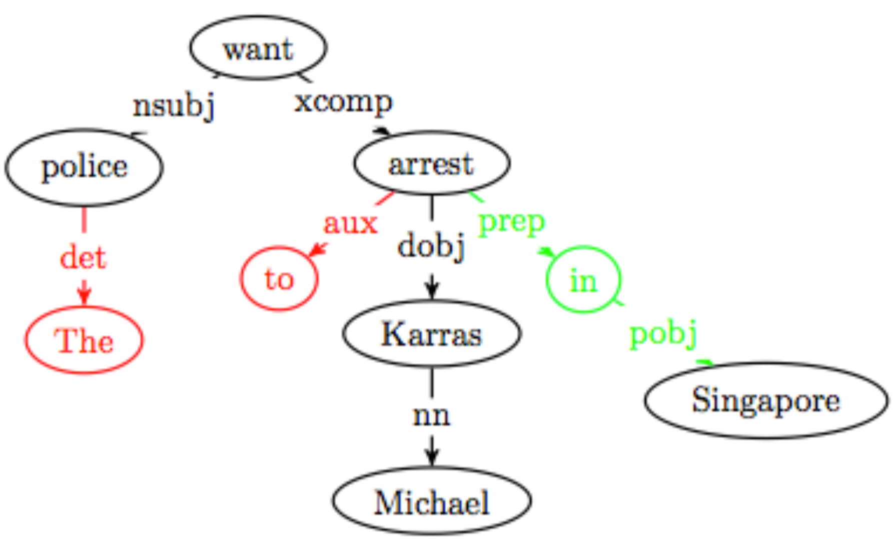
Results

	ACL 2014	Now
Full System (gold concepts)	80% Smatch	81% Smatch
Full System	58% Smatch	62% Smatch

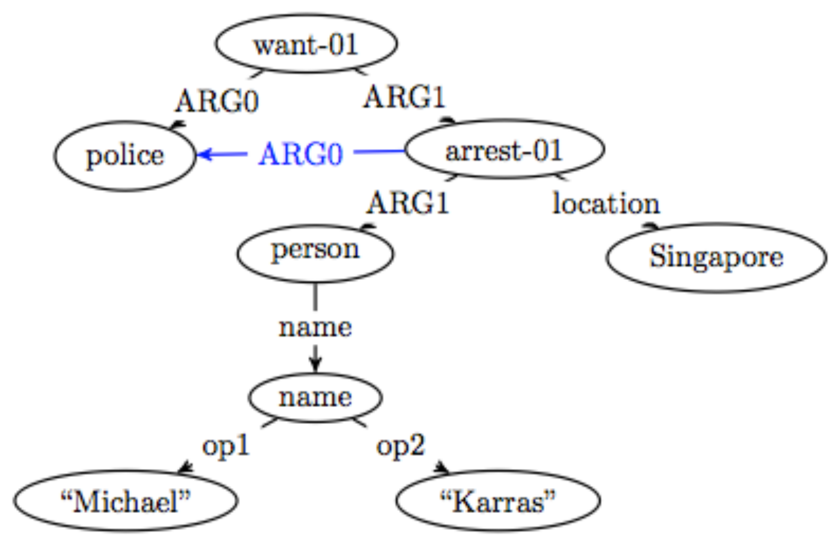
Transition-based AMR Parsing (Wang et al, NAACL 2015)

- Convert dependency tree into AMR graph
- Motivation: only a few difference between syntactic dependencies and AMR

Dependency tree



AMR graph

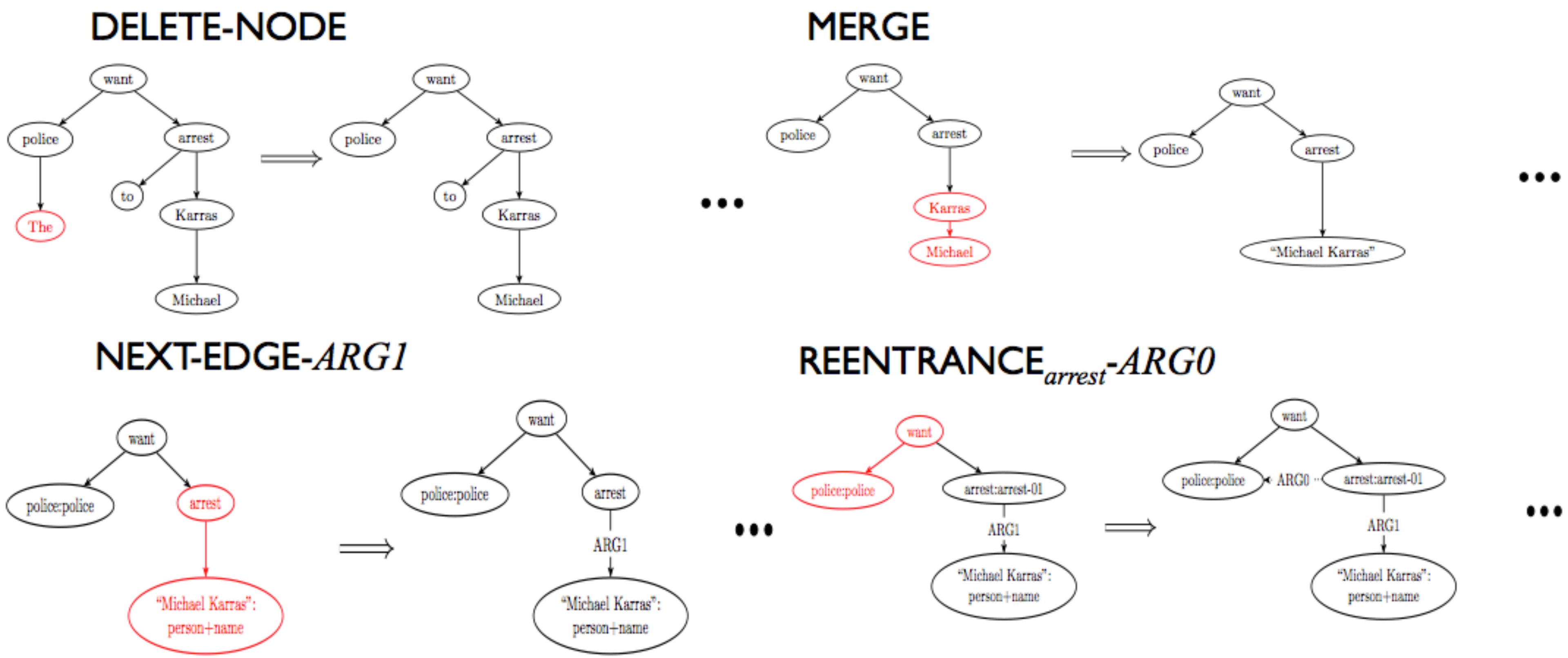


Transition-based Parsing

Transition-based AMR Parsing

- Parser transitions
  - **NEXT-EDGE-l_r** (attach edge and move to next word)
  - **SWAP-l_r** (swap nodes and attach with edge)
  - **REATTACH$_k$-l_r** (delete edge and reattach to already processed node)
  - **REPLACE-HEAD** (replace node with another node)
  - **REENTRANCE$_k$-l_r** (attach edge to already processed node)
  - **MERGE** (merge two nodes)
  - **NEXT-NODE-l_c** (label with concept and move to next word)
  - **DELETE-NODE** (deletes a word)

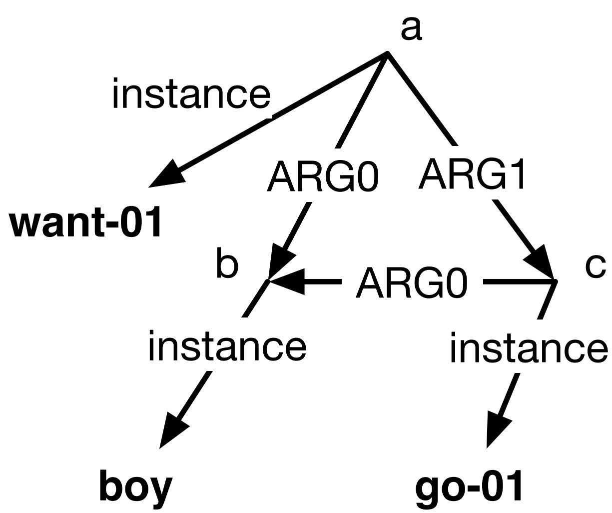
Transition-based AMR Parsing



Evaluation

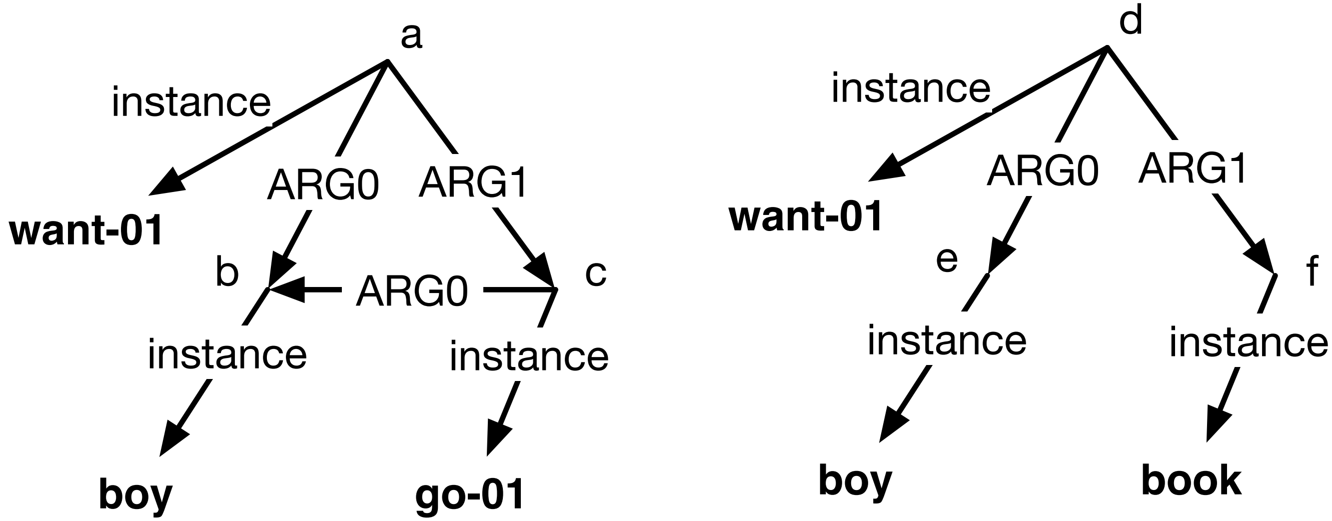
- Alignment
- Parsing
- **Evaluation**
- Graph Grammars and Automata
- Applications

Evaluation: Smatch (Cai & Knight, 2013)



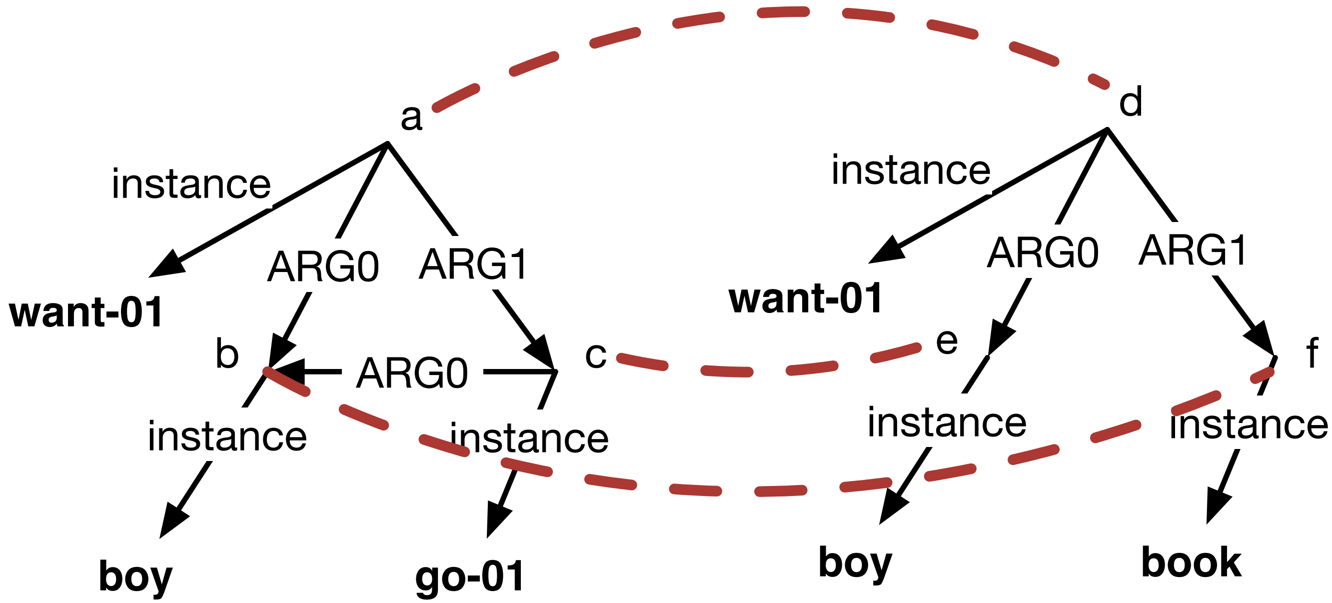
AMR graph w/ explicit instance edges

Evaluation: Smatch (Cai & Knight, 2013)



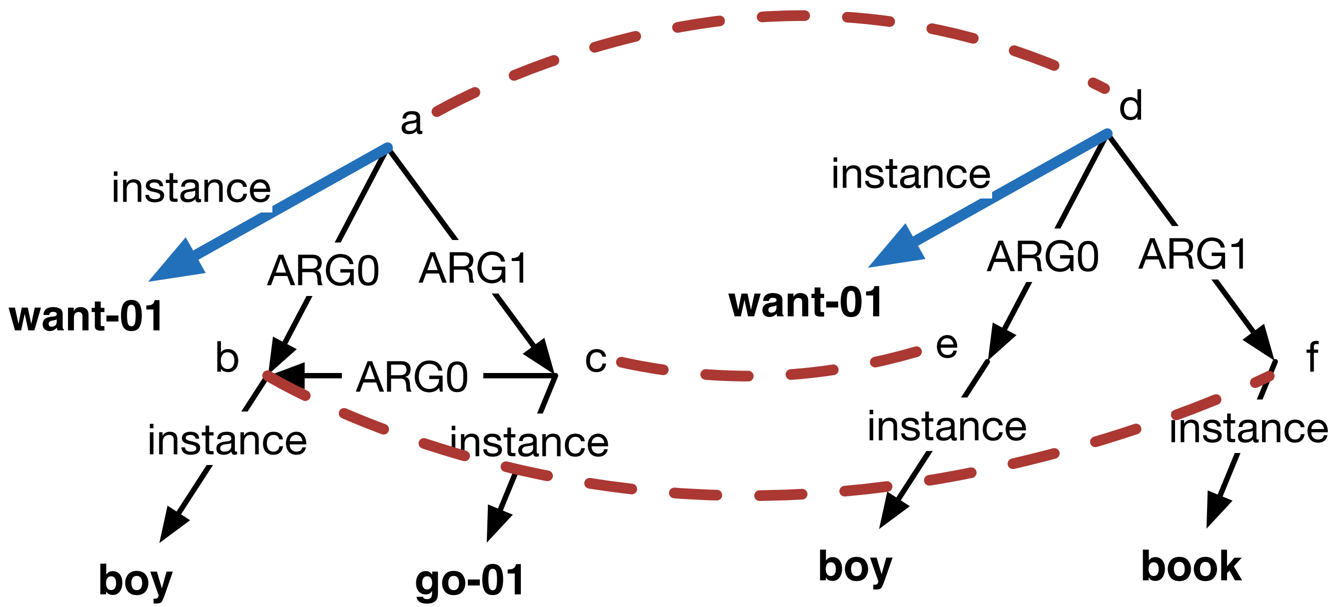
Want a number which indicates the similarity between two graphs

Evaluation: Smatch



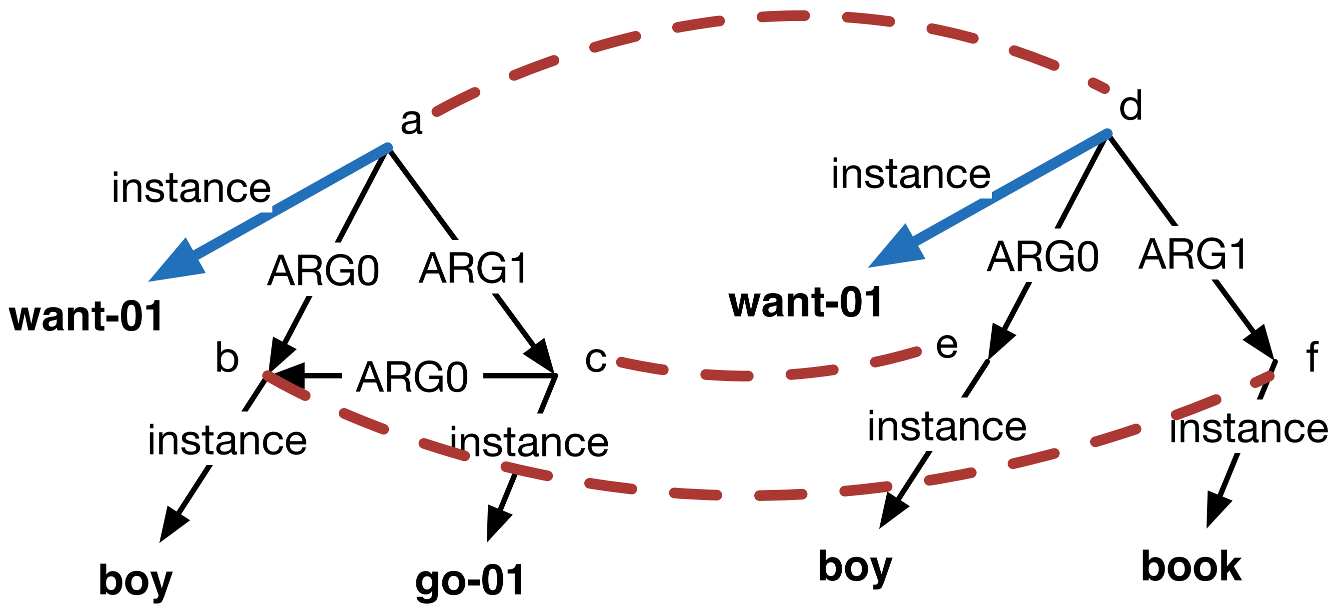
Consider an alignment between the nodes

Evaluation: Smatch



f-score = F_1 of identical matching edges
= 2 Match/($Total_1$ + $Total_2$)
= 2 / (6 + 5) = .18

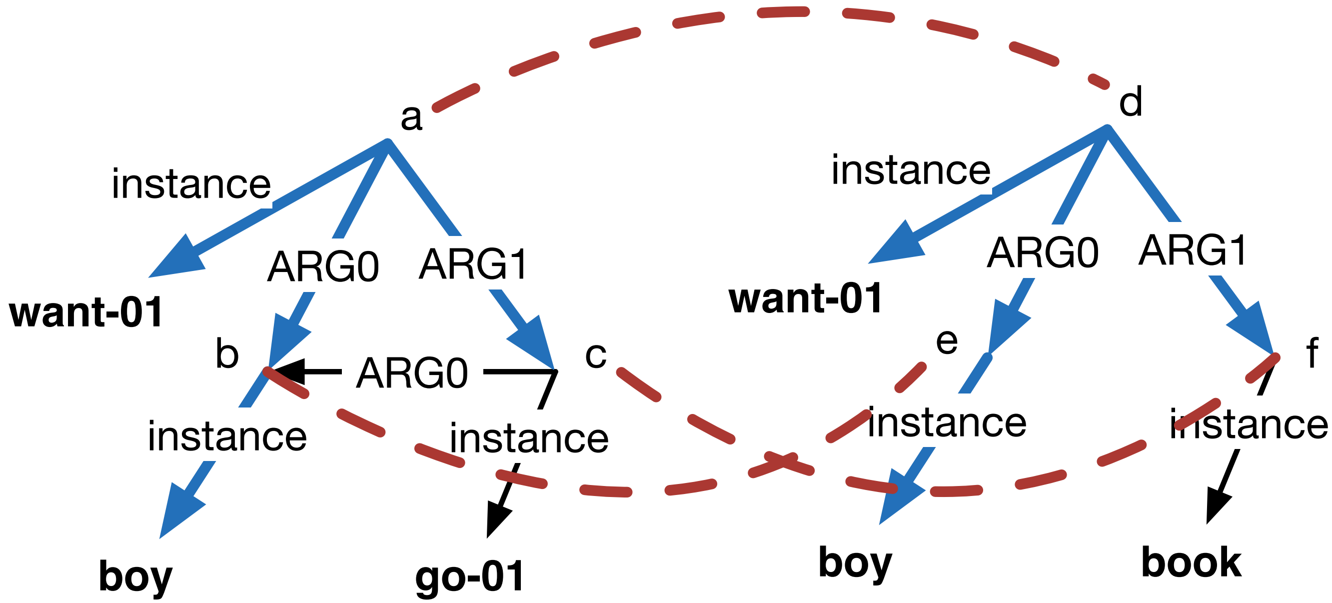
Evaluation: Smatch



Smatch score = maximum f-score over all possible alignments

NP hard => approximate inference to find highest scoring alignment

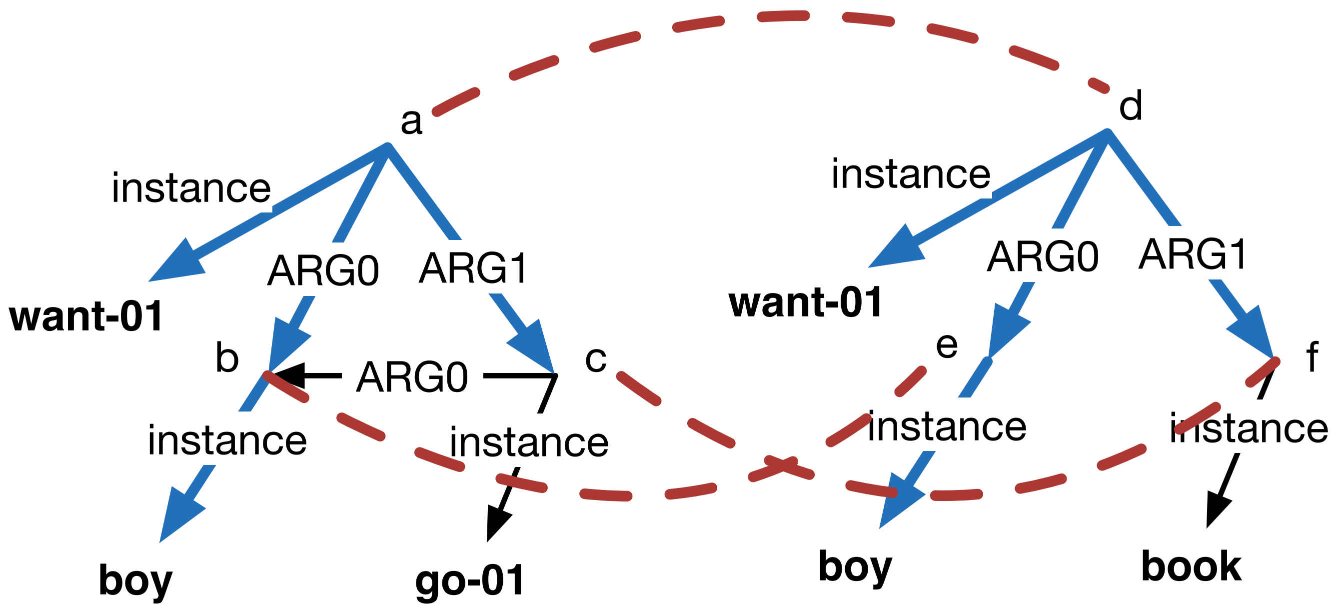
Evaluation: Smatch



Smatch score = 8 / (6 + 5) = .73

Highest scoring alignment

Evaluation: Smatch



Multi-lingual version of Smatch: demo by Naomi Saphra at NAACL 2015

Roadmap

- Evaluation
- Alignment
- Parsing
- **Graph Grammars and Automata**
  - Background: CFGs and tree substitution grammars
  - Hyperedge Replacement Grammars (HRGs)
  - Directed Acyclic Graph (DAG) Automata
- Applications

Motivation for Graph Grammars

- String and tree grammars, automata, transducers, etc widely used in NLP applications
  - Phrase structure parsers, syntactic MT systems
- Semantics (like AMR) is represented as graphs

We would like grammars, automata, transducers, etc over graphs

Context Free Grammar (CFG)

Grammar

1) S -> NP VP
2) NP -> We
3) VP -> V NP
4) V -> want
5) V -> like
6) NP -> ice cream

Context Free Grammar (CFG)

Grammar

1) S -> NP VP
2) NP -> We
3) VP -> V NP
4) V -> want
5) V -> like
6) NP -> ice cream

Example derivation

S

Context Free Grammar (CFG)

<u>Grammar</u>

1) S -> NP VP
2) NP -> We
3) VP -> V NP
4) V -> want
5) V -> like
6) NP -> ice cream

<u>Example derivation</u>

S

$\Rightarrow_1$ NP VP

Context Free Grammar (CFG)

Grammar

1) S -> NP VP
2) NP -> We
3) VP -> V NP
4) V -> want
5) V -> like
6) NP -> ice cream

Example derivation

S

$\Rightarrow_1$ NP VP

$\Rightarrow_3$ NP V NP

Context Free Grammar (CFG)

Grammar

1) S -> NP VP
2) NP -> We
3) VP -> V NP
4) V -> want
5) V -> like
6) NP -> ice cream

Example derivation

S

$\Rightarrow_1$ NP VP

$\Rightarrow_3$ NP V NP

$\Rightarrow_4$ NP like NP

Context Free Grammar (CFG)

Grammar

1) S -> NP VP
2) NP -> We
3) VP -> V NP
4) V -> want
5) V -> like
6) NP -> ice cream

Example derivation

S

$\Rightarrow_1$ NP VP

$\Rightarrow_3$ NP V NP

$\Rightarrow_4$ NP like NP

$\Rightarrow_6$ NP like ice cream

Context Free Grammar (CFG)

Grammar

1) S -> NP VP
2) NP -> We
3) VP -> V NP
4) V -> want
5) V -> like
6) NP -> ice cream

Example derivation

S

$\Rightarrow_1$ NP VP

$\Rightarrow_3$ NP V NP

$\Rightarrow_4$ NP like NP

$\Rightarrow_6$ NP like ice cream

$\Rightarrow_2$ We like ice cream

Context Free Grammar (CFG)

Grammar

1) S -> NP VP
2) NP -> We
3) VP -> V NP
4) V -> want
5) V -> like
6) NP -> ice cream

Example derivation

$$
\begin{aligned}
& S \\
\Rightarrow_1 \; & NP\ VP \\
\Rightarrow_3 \; & NP\ V\ NP \\
\Rightarrow_4 \; & NP\ \text{like}\ NP \\
\Rightarrow_6 \; & NP\ \text{like ice cream} \\
\Rightarrow_2 \; & \underbrace{\text{We like ice cream}}_{\text{Yield}}
\end{aligned}
$$

Context Free Grammar (CFG)

<u>Grammar</u>

1) S -> NP VP
2) NP -> We
3) VP -> V NP
4) V -> want
5) V -> like
6) NP -> ice cream

<u>Example derivation</u>

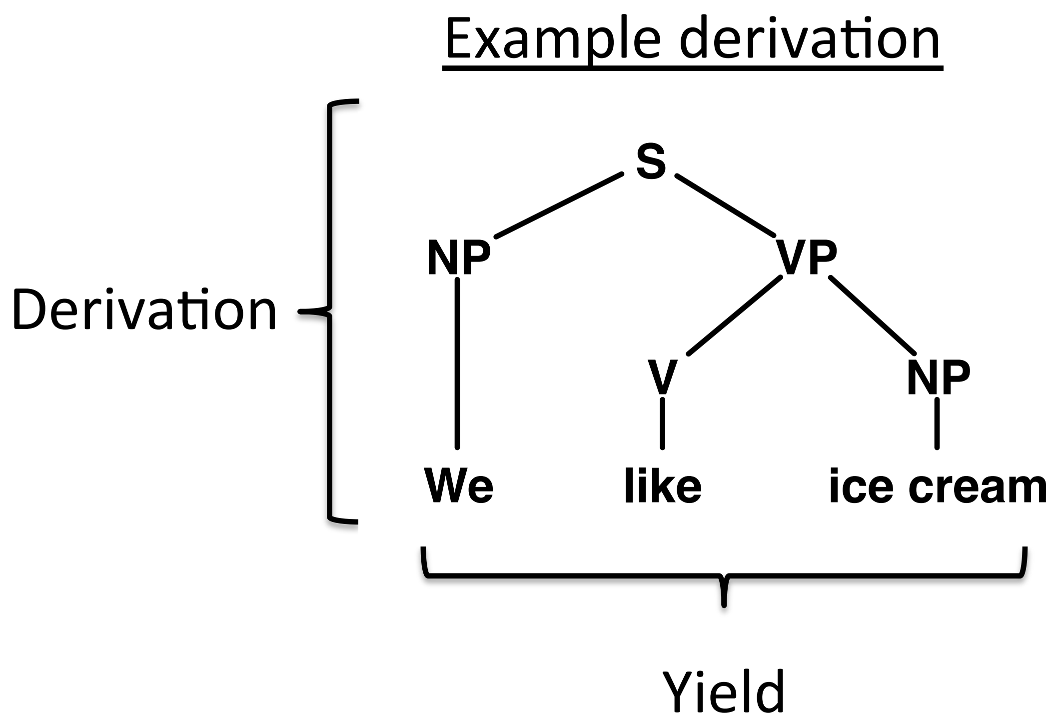
Derivation

$$
\left.
\begin{array}{l}
\quad \text{S} \\
\Rightarrow_1 \text{NP VP} \\
\Rightarrow_3 \text{NP V NP} \\
\Rightarrow_4 \text{NP like NP} \\
\Rightarrow_6 \text{NP like ice cream} \\
\Rightarrow_2 \underbrace{\text{We like ice cream}}_{\text{Yield}}
\end{array}
\right.
$$

Context Free Grammar (CFG)

Grammar

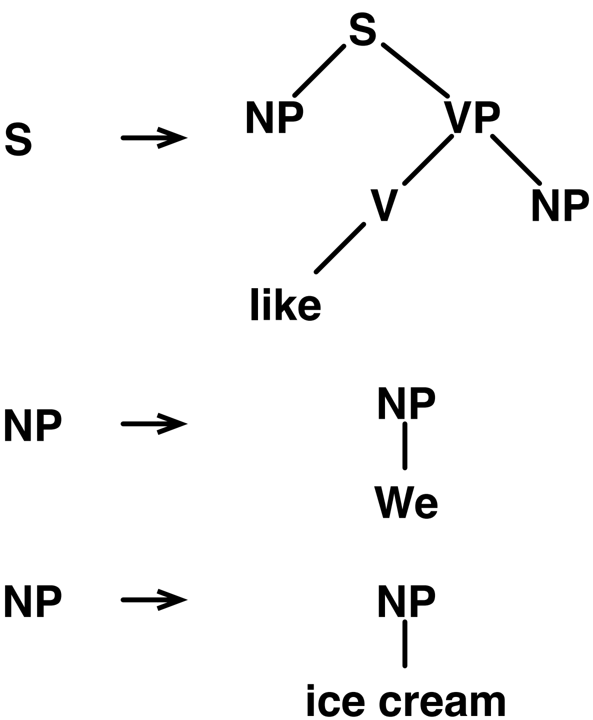
1) S -> NP VP
2) NP -> We
3) VP -> V NP
4) V -> want
5) V -> like
6) NP -> ice cream



Productions RHS are **strings,** derivations are **trees**

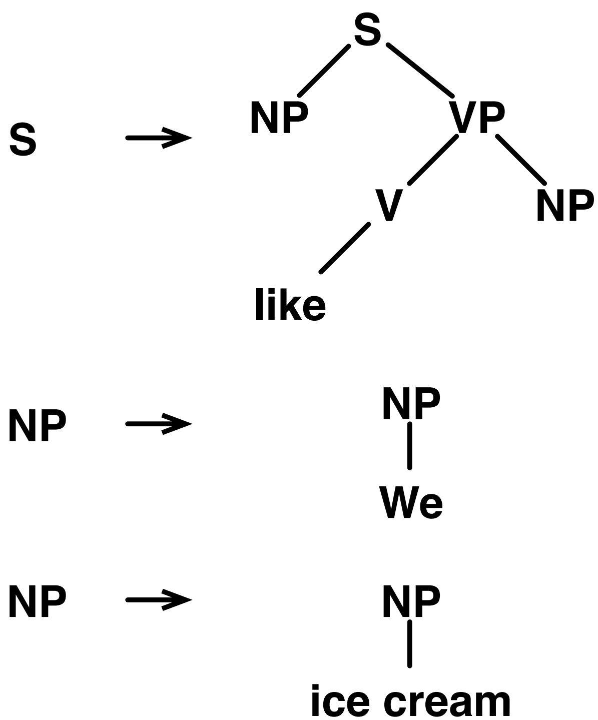
Tree Substitution Grammar (TSG)

Grammar



Tree Substitution Grammar (TSG)

Grammar



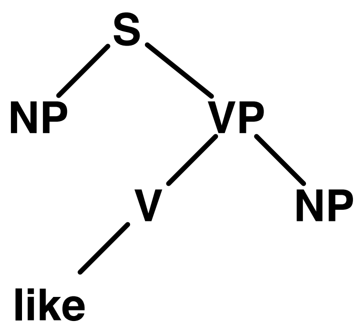
Example derivation

S

Tree Substitution Grammar (TSG)

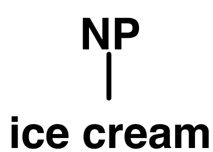
Grammar

S →



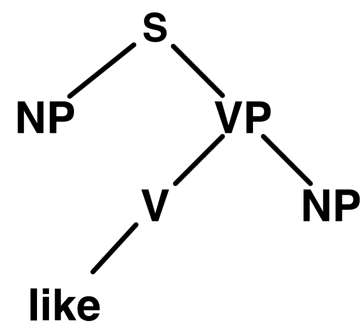
NP →

NP →



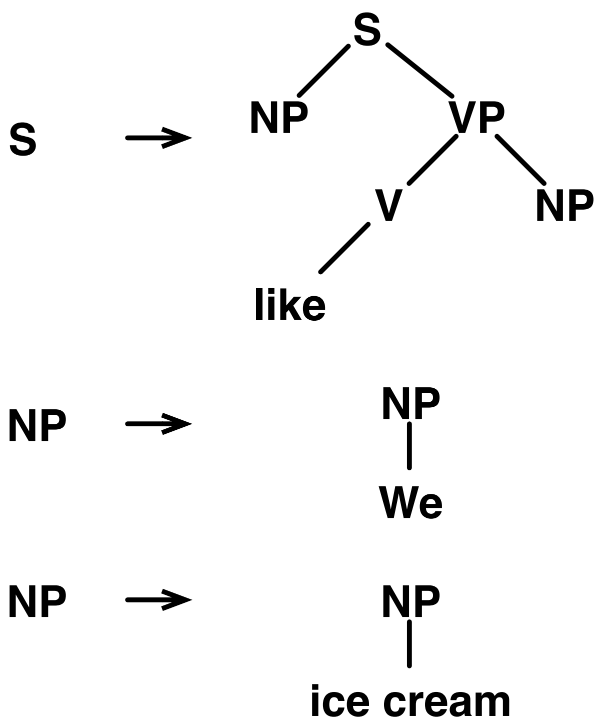
Example derivation

S ⇒

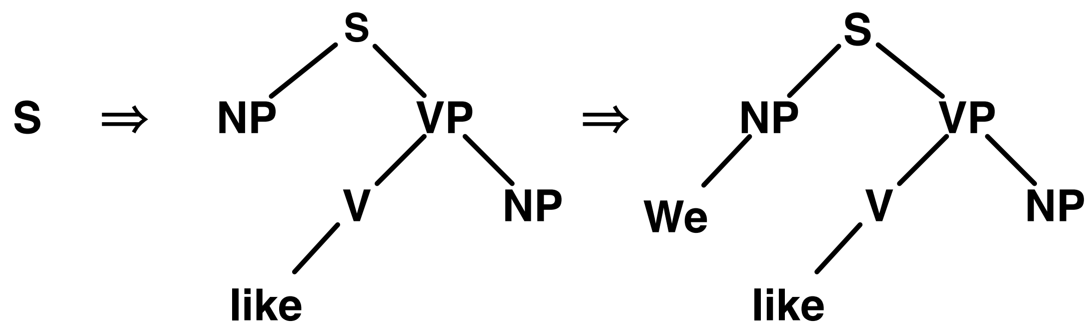


Tree Substitution Grammar (TSG)

Grammar

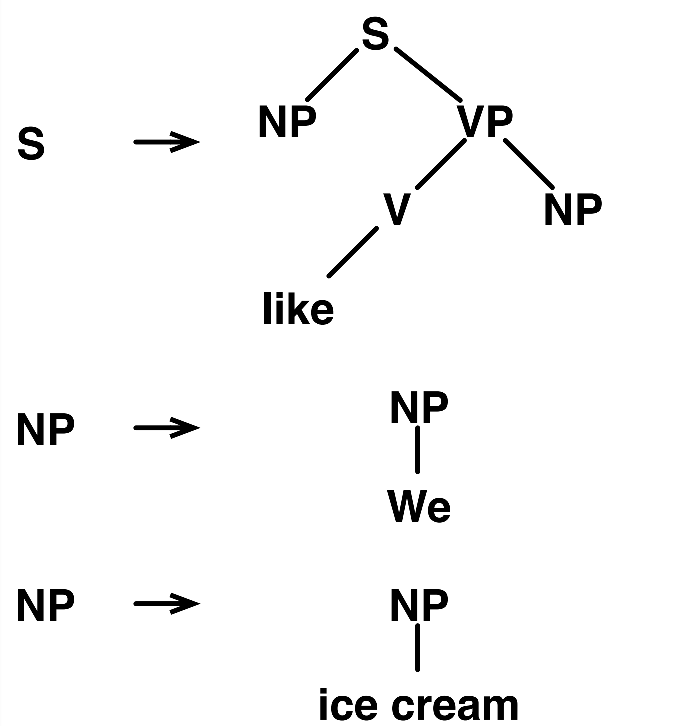


Example derivation

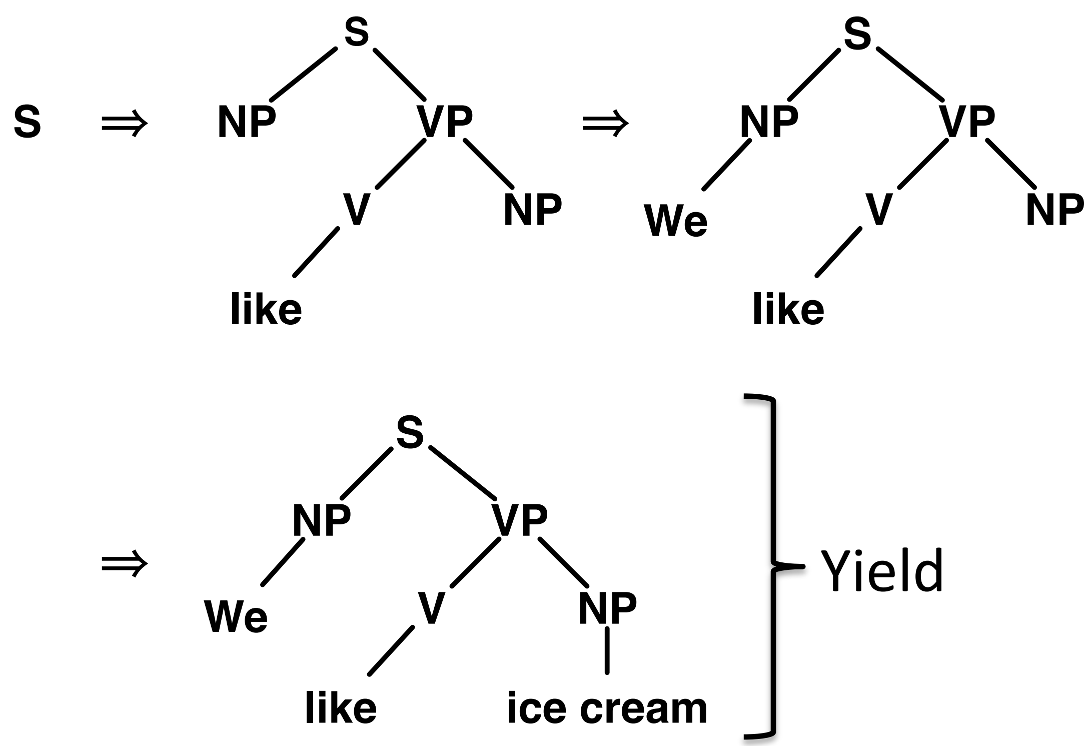


Tree Substitution Grammar (TSG)

Grammar

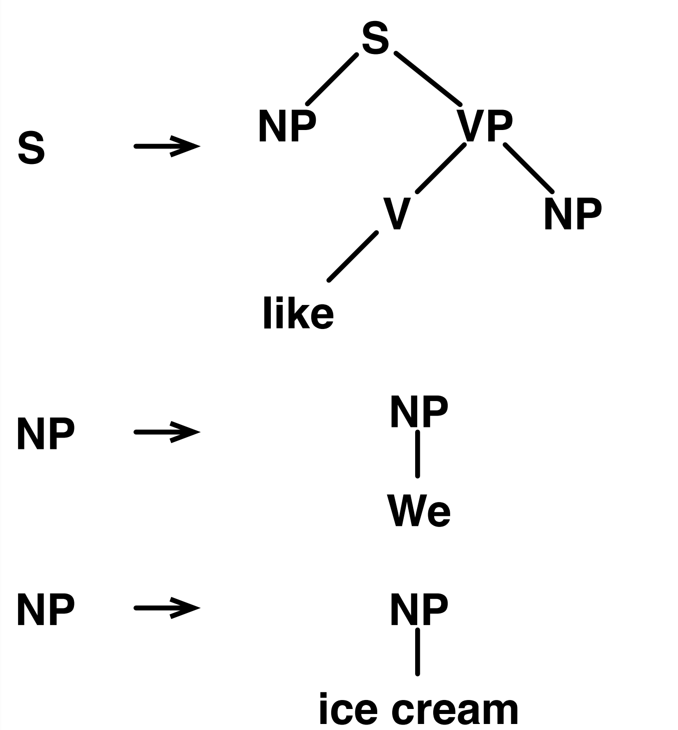


Example derivation

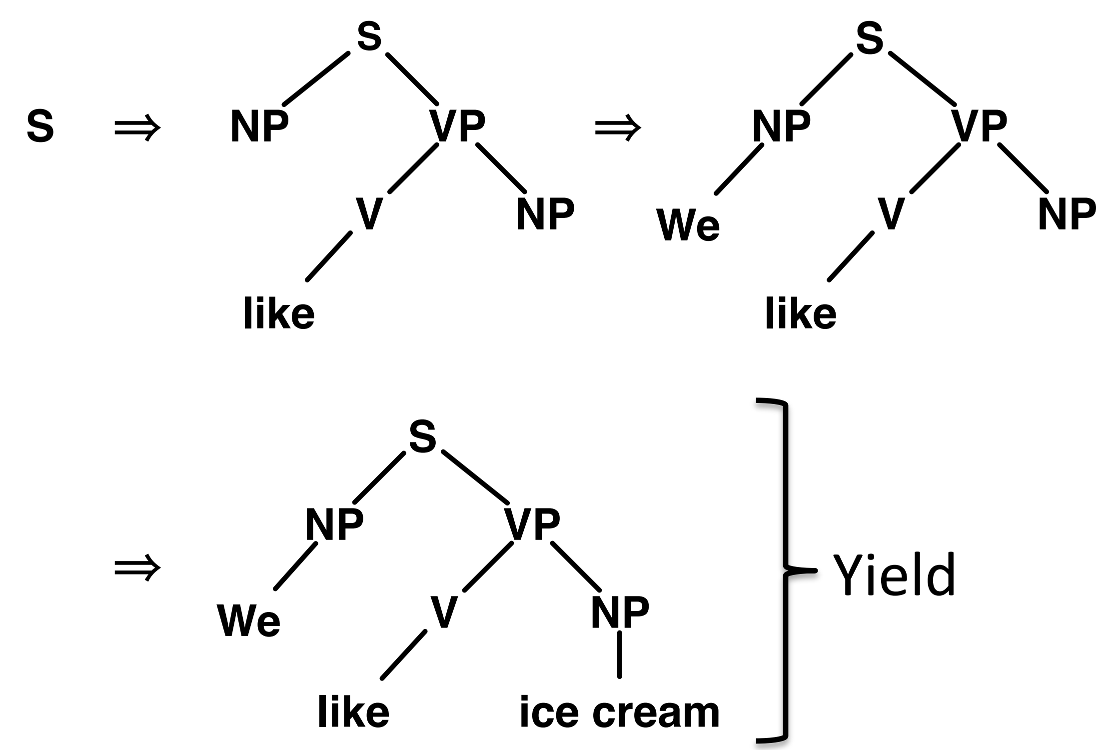


Tree Substitution Grammar (TSG)

Grammar



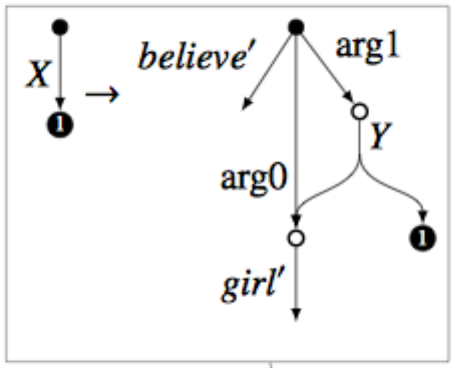
Example derivation

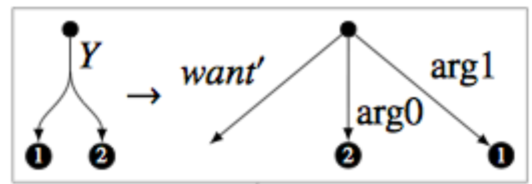


Productions RHS are **trees,** derivations are **trees**

Hyperedge Replacement Grammar (HRG)

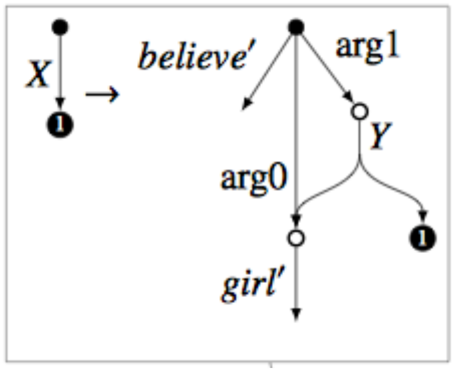
Grammar rules (some of them)

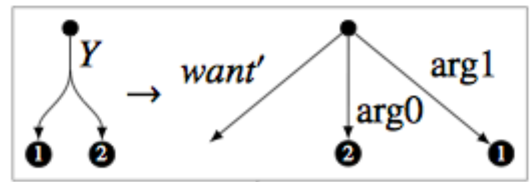


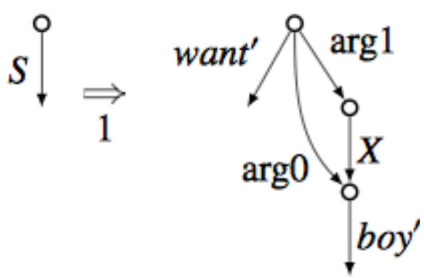


Hyperedge Replacement Grammar (HRG)

<u>Grammar rules (some of them)</u>

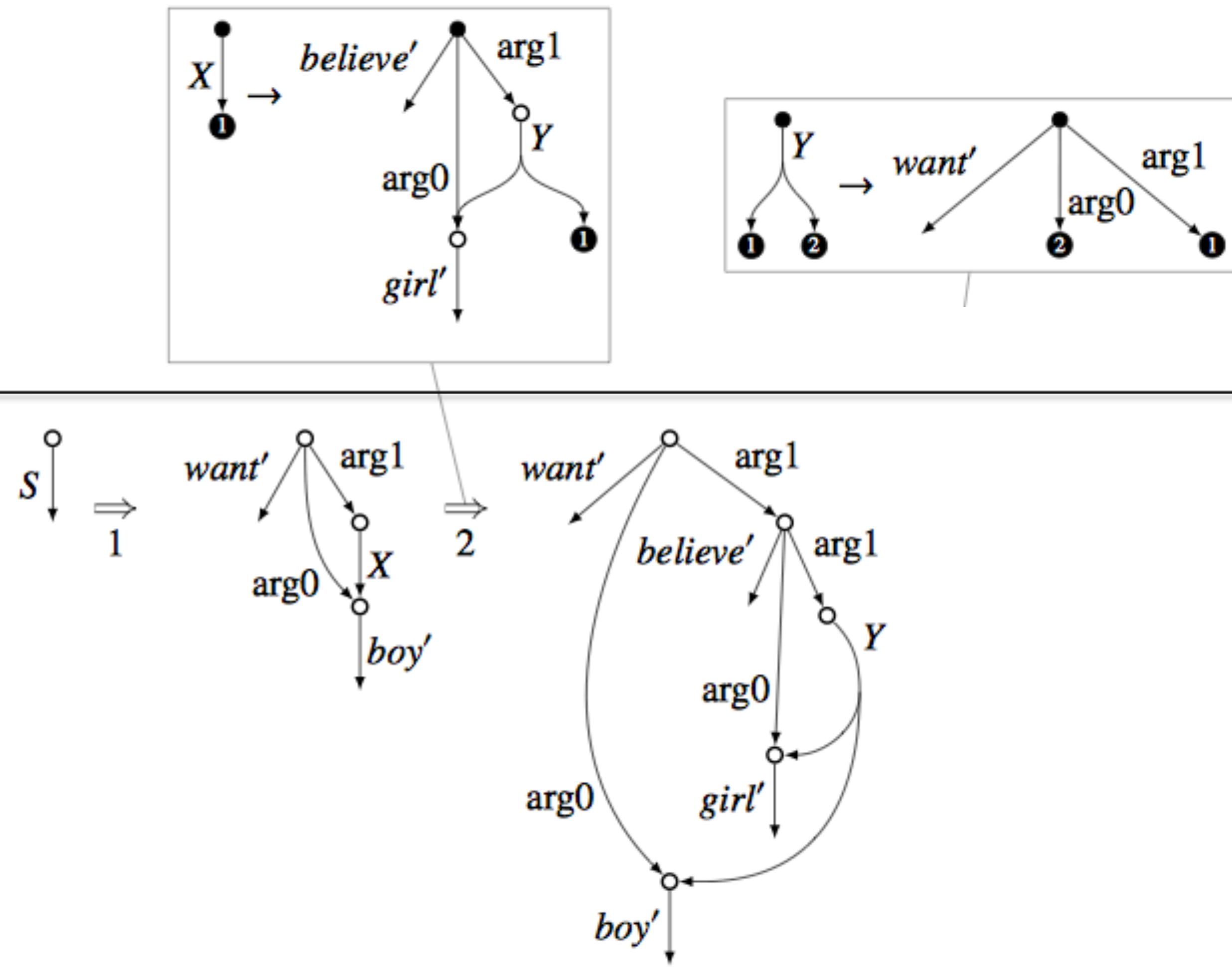






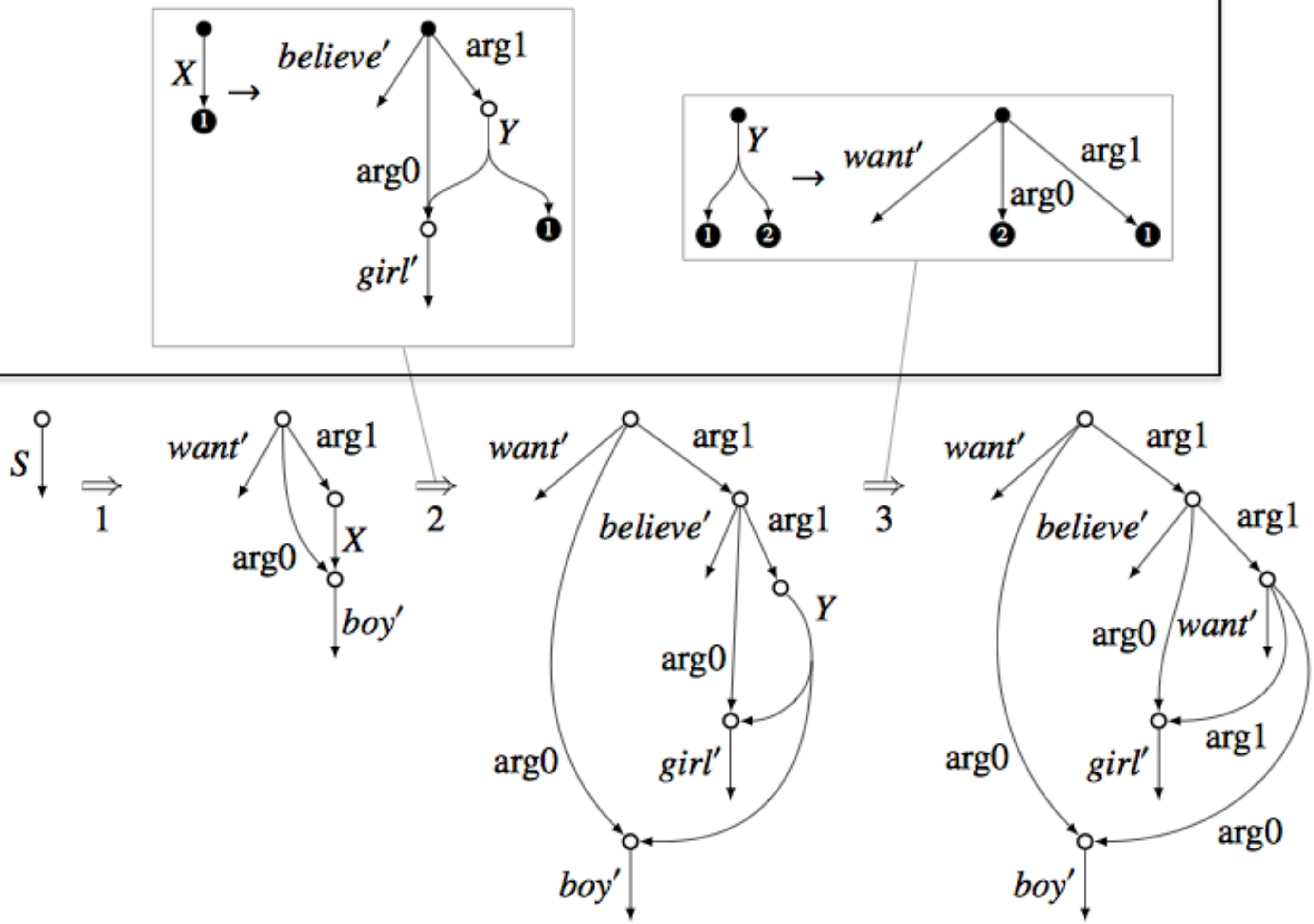
Hyperedge Replacement Grammar (HRG)

<u>Grammar rules (some of them)</u>

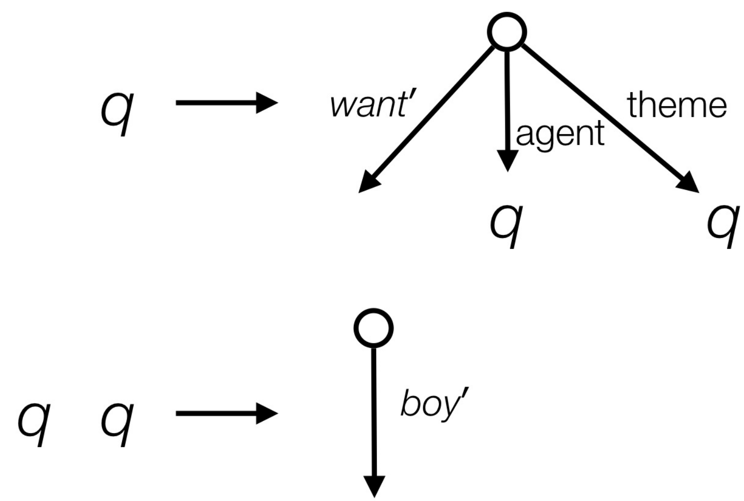


Hyperedge Replacement Grammar (HRG)

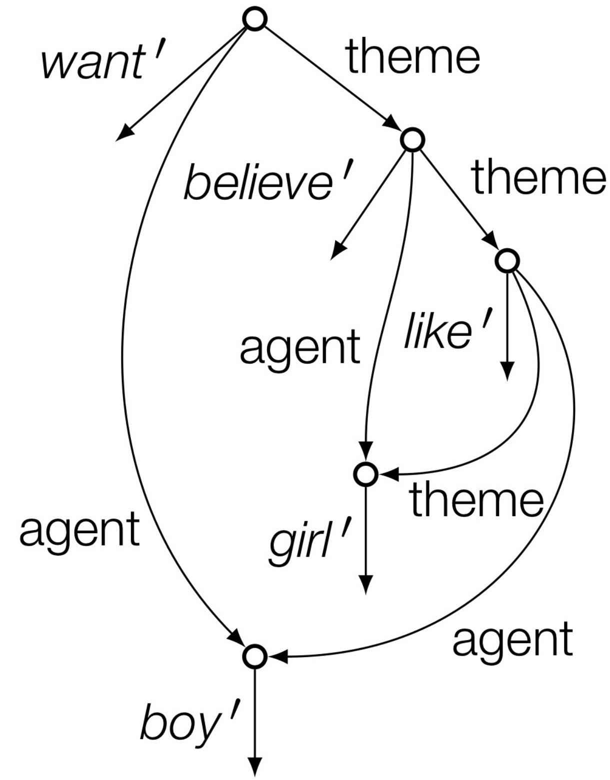
Grammar rules (some of them)



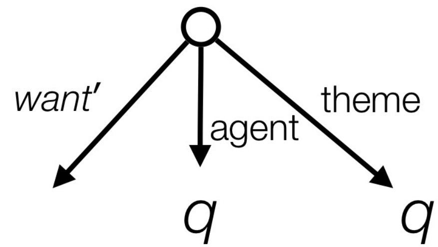
DAG automata (Kamimura and Slutzki, 1981)



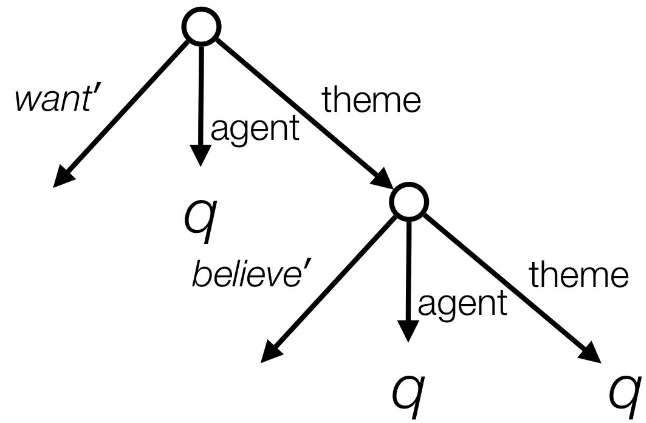
Two or more states can merge



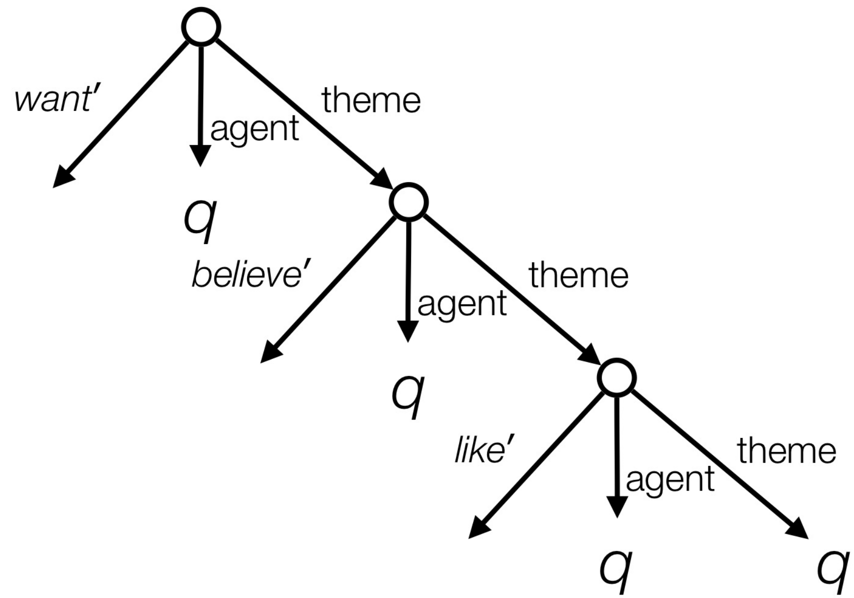
DAG automata (Kamimura and Slutzki, 1981)



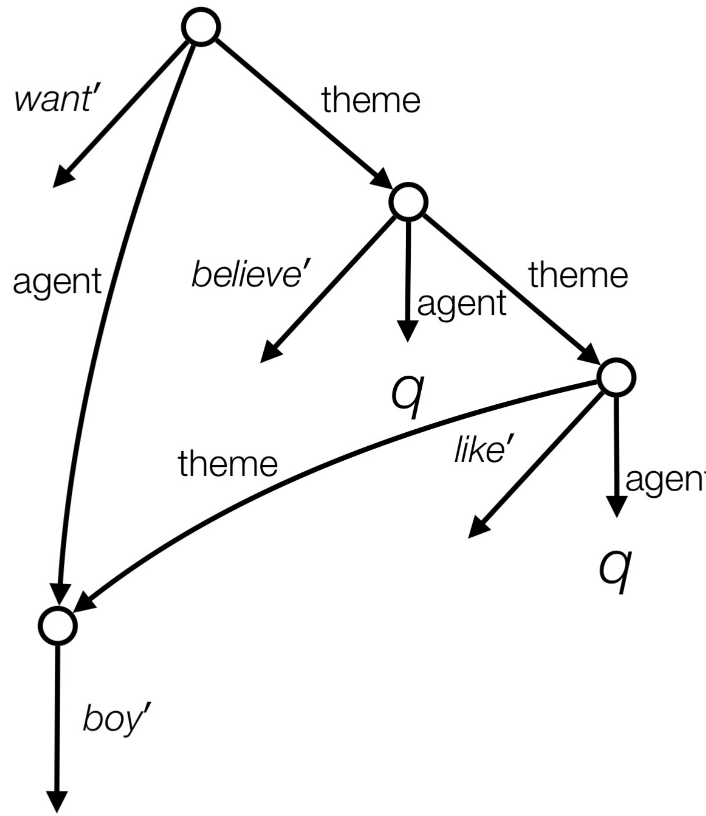
DAG automata (Kamimura and Slutzki, 1981)



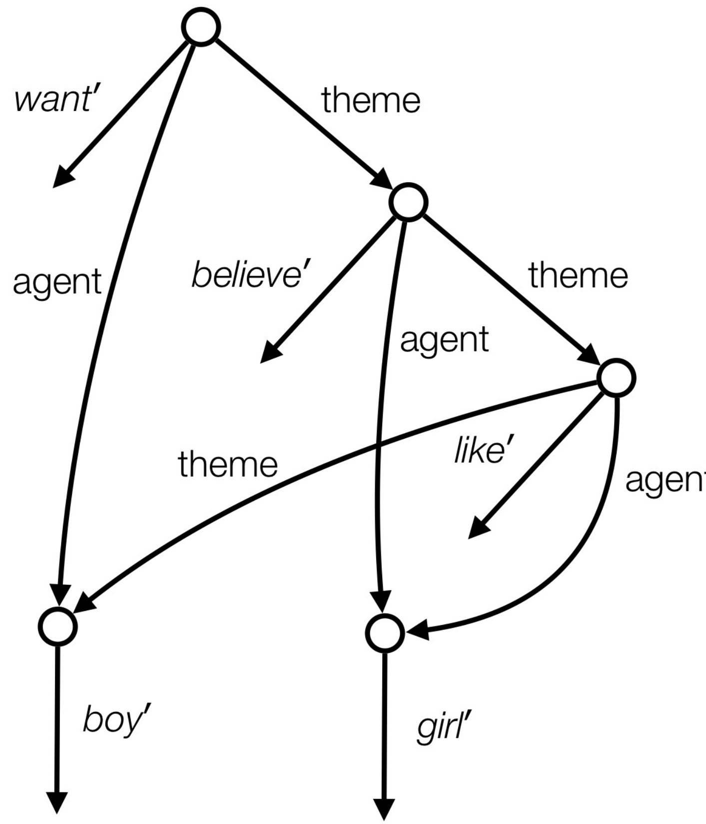
DAG automata (Kamimura and Slutzki, 1981)



DAG automata (Kamimura and Slutzki, 1981)



DAG automata (Kamimura and Slutzki, 1981)



Extensions

- Weighted and probabilistic grammars
- Synchronous grammars and transducers
  - Useful for building parsers, generators, and MT systems

Recent/Ongoing work

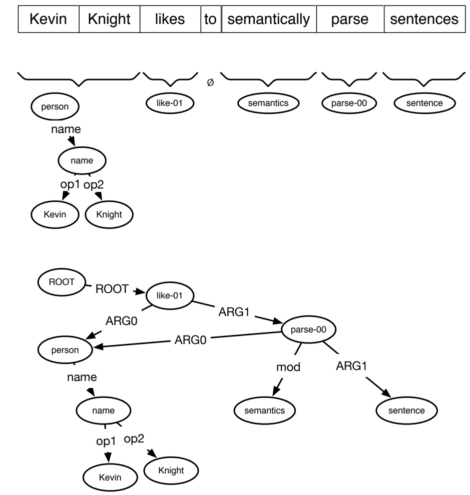
- Improved parsing algorithms (Chaing et al, 2013)
- Applications to parsing and generation (Braune et al, 2014) and MT (Jones et al, 2012)
- Implementations
  - Hyperedge replacement grammars: **Bolinas** (Chaing et al, 2013; Jones et al, 2012)
  - DAG automata: **DAGGER** (Quernheim & Knight, 2012)

Alignment

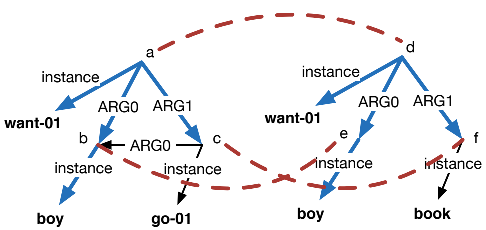
IAEA accepted North Korea 's proposal in November.

```
(a / accept-01
     :ARG0 (o / organization
          :ARG0 (n / name
               :op1 "IAEA"))
     :ARG1 (t2 / thing
          :ARG1-of (p / propose-01
               :ARG0 (c / country
                    :name (n2 / name
                         :op1 "North"
                         :op2 "Korea"))))
     :time (d / date-entity
               :month 11))
```

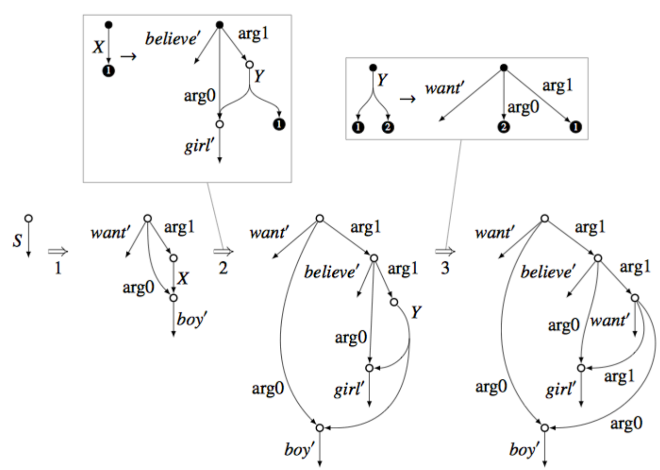
Parsing



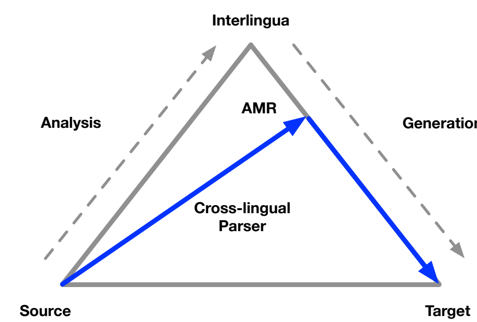
Evaluation



Graph grammars



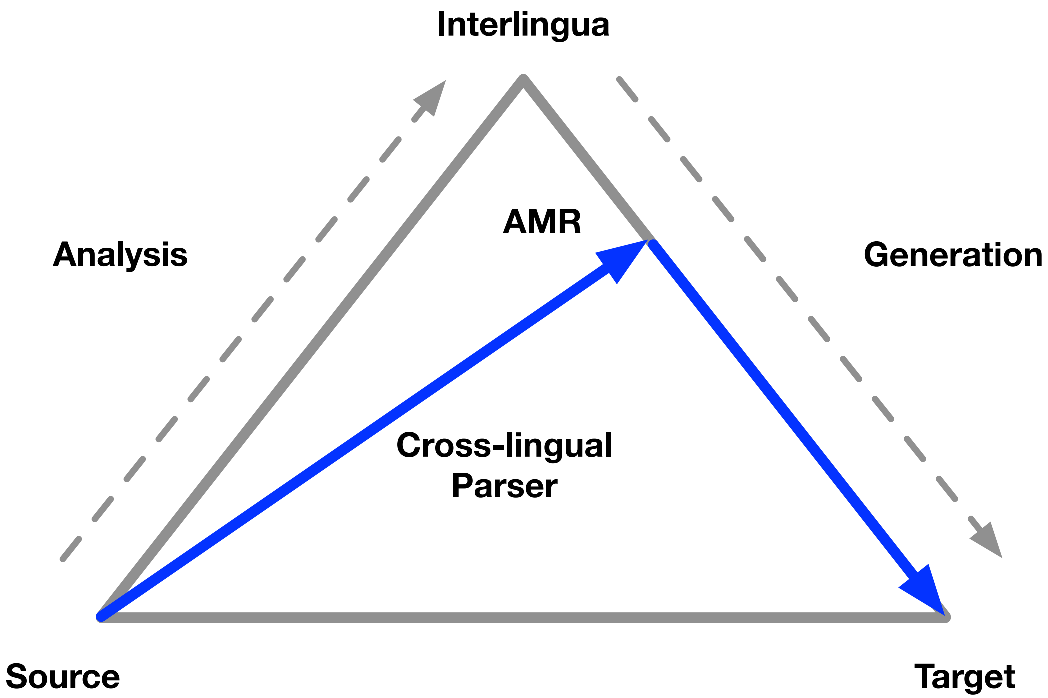
Applications



Applications

- Alignment
- Parsing
- Evaluation
- Graph Grammars and Automata
- **Applications**
  - MT, Summarization, Entity linking

Machine Translation



Summarization

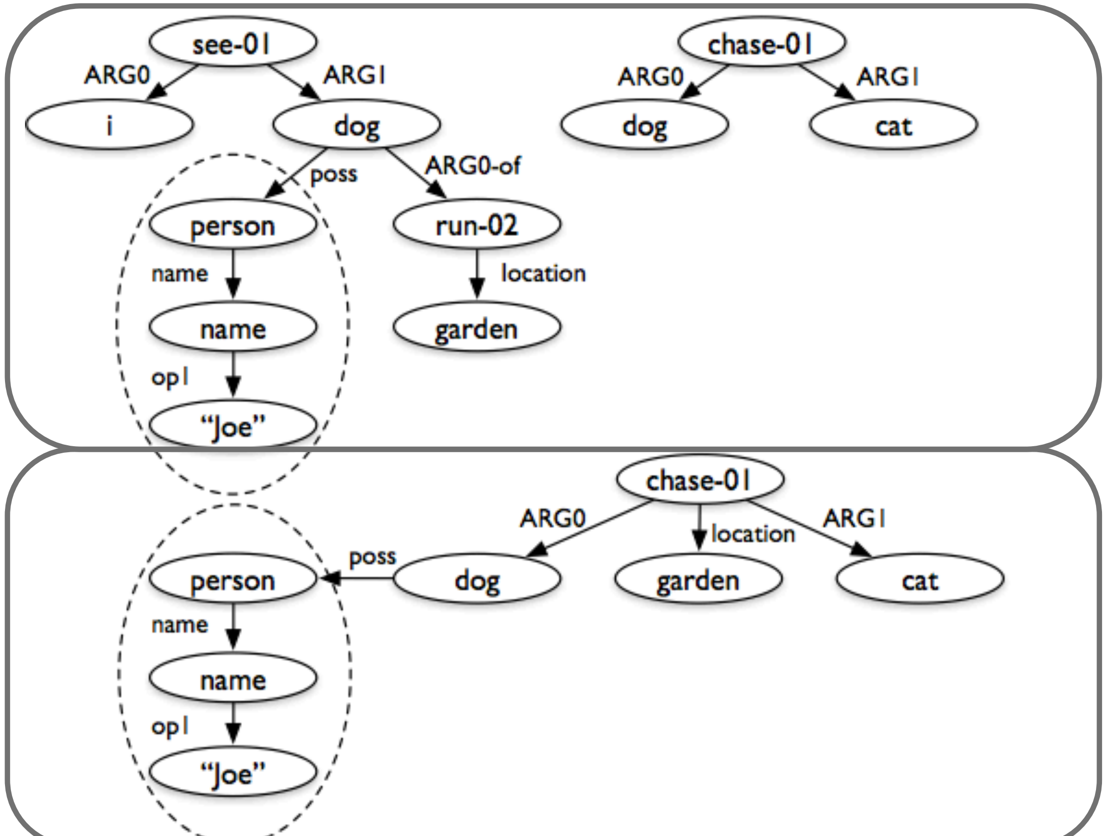
Document Sentences

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

Sentence B: The dog was chasing a cat.

Document AMRs

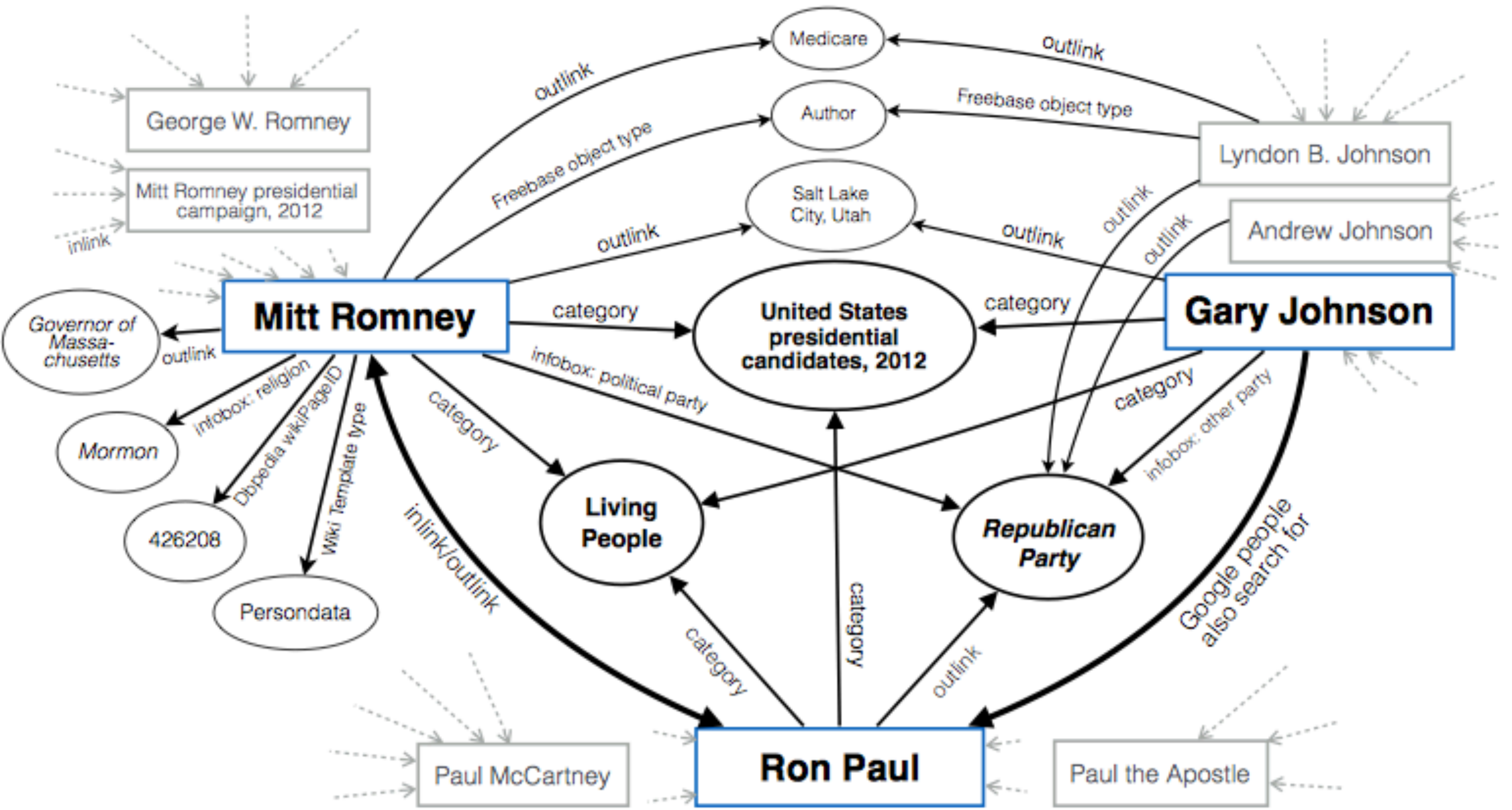
Summary AMR (select nodes and edges)



Summary

Summary: Joe's dog was chasing a cat in the garden.

Entity Linking



AMR at NAACL 2015

acknowledgements

```
(t / thank-01
     :ARG1 (y / you))
```

Links, etc

Resources

- AMR website:
- JAMR:
- Transition-based parser: https://github.com/Juicechuan/AMRParsing/tree/v2
- Bolinas
- DAGGER

References

Backup slides

Structured Prediction

- Define input space and **output space**
- Decide **scope of the features**
- **Model** type (linear, neural, etc)
- Come up with an **inference algorithm** (dynamic programming, integer linear programming, etc)
- Pick a **learning algorithm,** usually an objective + optimizer (conditional log likelihood + stochastic gradient descent, perceptron loss + Adagrad, etc)

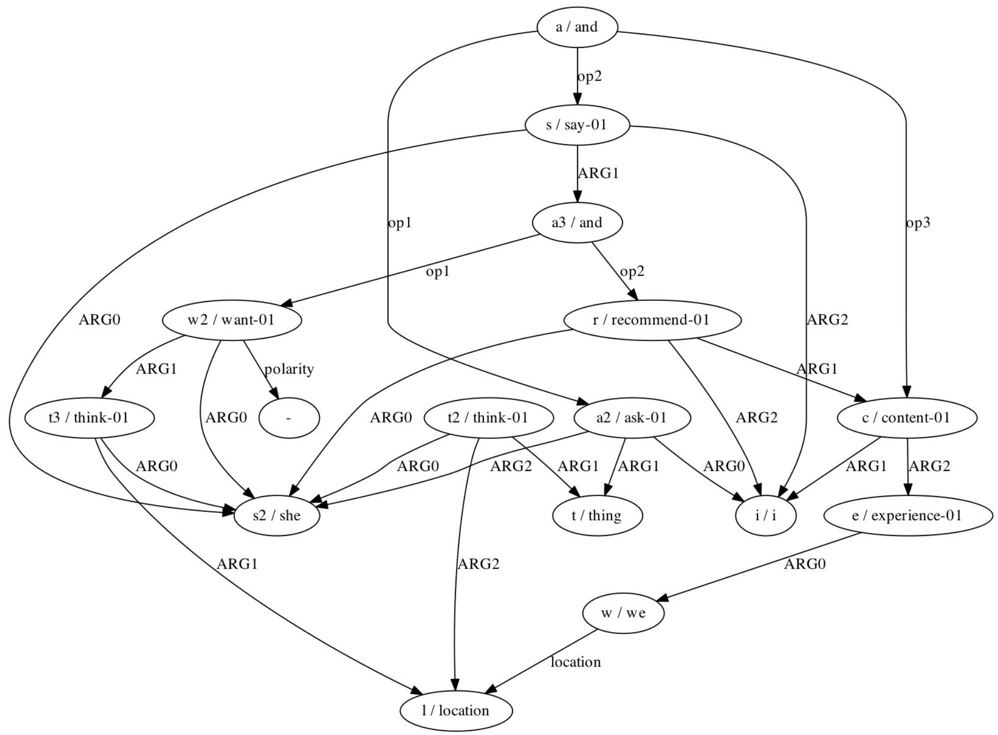
Structured Prediction

	Concept ID	**Relation ID**
Input	Sentence	Sentence labeled w/ graph fragments
Output	Sentence labeled w/ graph fragments	Connected deterministic edge-labeled directed graph spanning the input fragments
Model	Linear	Linear
Scope of features	Zeroth order semi-Markov (considers label of each span in isolation)	Edge local (looks at edge label and head and tail nodes)
Inference algorithm (finds best output)	Dynamic programming	Graph algorithm + Lagrangian relaxation
Learning algorithm	Perceptron loss w/ Adagrad	Perceptron loss w/ Adagrad

Outline (detailed)

- Evaluation: Smatch
- Alignment
  - JAMR's rule-based aligner
  - Alignment w/ EM
- Parsing
  - Graph-based parsing
    - Concept identification
    - Relation identification
      - Maximum spanning connected graph algorithm (MSCG)
      - Graph determinism constraints using Lagrangian relaxation
    - Experiments
  - Transition-based parsing
- Graph Formalisms
  - DAG Automata
  - Hyper-edge Replacement Grammars (HRGs)
- Applications
  - MT, Summarization, Entity linking

Real-world treewidth



k=4

I asked her what she thought about where we'd be and she said she doesn't want to think about that, and that I should be happy about the experiences we've had (which I am).

Deleted slides

Intro

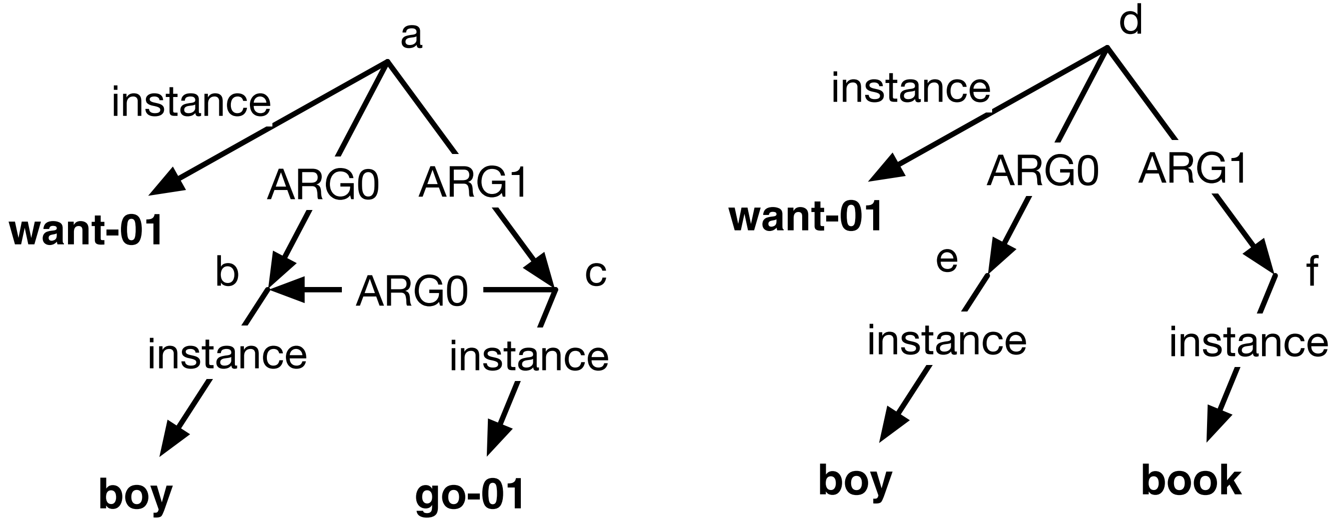
- Going from natural language to AMR is parsing
- Going from generation to AMR generation

Smatch

- (Picture of two AMRs)
- (Next slide: show alignment and things that match in black, things that don't match in grey)
- Smatch is defined as highest Smatch score from all possible alignments
- NP hard => approximate inference to find best alignment

Cai & Knight (2013)

Evaluation: Smatch (Cai & Knight, 2013)



The boy wants to go. The boy wants the book.

Alignment

The tour was a surprise offer made by North Korea in November.

```
(t / thing
    :ARG0-of (s / surprise-01)
    :ARG1-of (o / offer-01
        :ARG0 (c / country
            :name (n / name
                :op1 "North"
                :op2 "Korea"))
        :time (d / date-entity
                :month 11))
    :domain (t2 / tour-01))
```

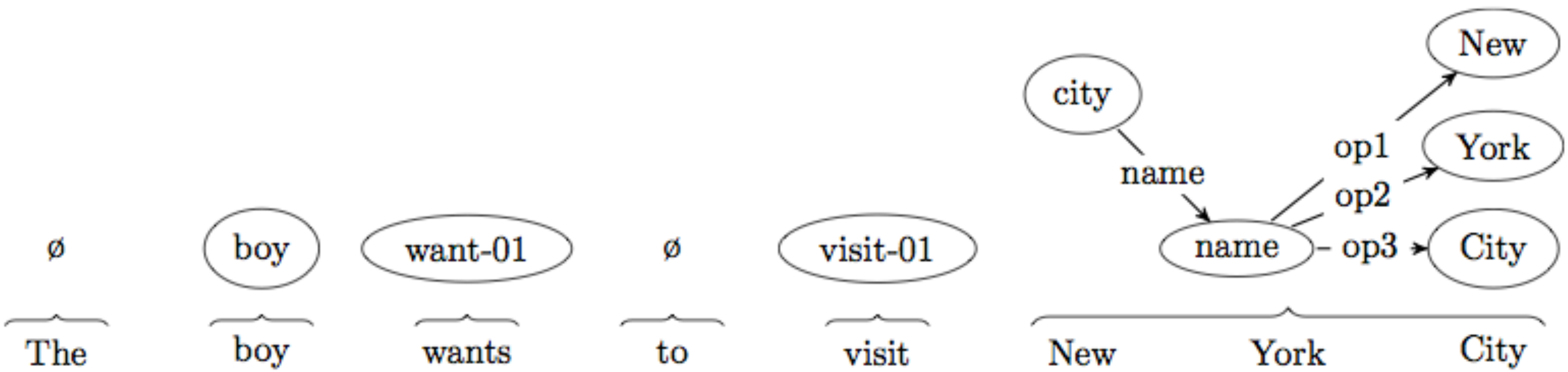
Alignment

The tour was a surprise offer made by North Korea in November.

```
(t / thing
     :ARG0-of (s / surprise-01)
     :ARG1-of (o / offer-01
          :ARG0 (c / country
               :name (n / name
                    :op1 "North"
                    :op2 "Korea"))
          :time (d / date-entity
                    :month 11))
     :domain (t2 / tour-01))
```

Align concepts with words

JAMR's Aligner



- For each concept in the AMR graph, it searches the sentence for corresponding span of words using a list of rules. It uses:
  - WordNet
  - Edit Distance

JAMR's Aligner

- Rules for:
  - Named entities, date entities
  - Single concepts matched by lemma and string edit distance
  - Person, organization, thing and have-org-role-91 concepts
  - Negation, quantities, degree

JAMR's Aligner

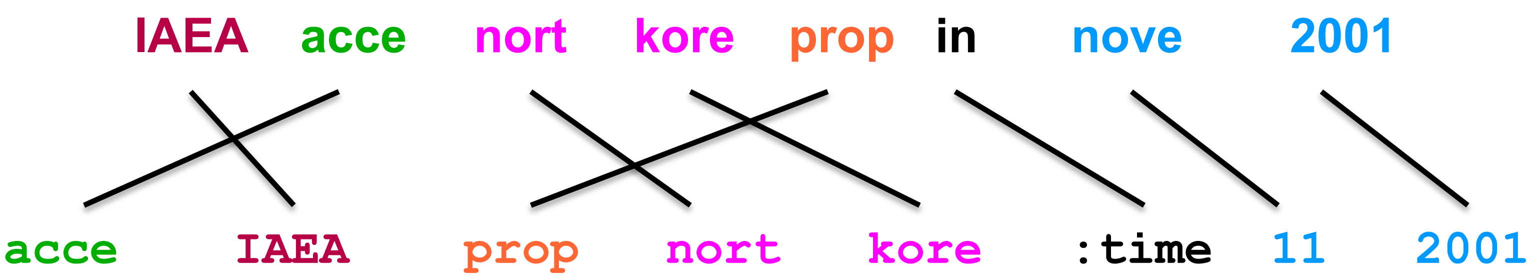
The tour was a surprise invitation made by North Korea in November 2001.

```
(t / thing
    :ARG0-of (s / surprise-01)
    :ARG2-of (i / invite-01
        :ARG0 (c / country
            :name (n / name
                :op1 "North"
                :op2 "Korea"))
        :time (d / date-entity
                :month 11
                :year 2001))
    :domain (t2 / tour-01))
```

JAMR’s Aligner

- Slide: the rules and how they work
- Slide: things it misses

EM Alignment



Run IBM alignment model 1, HMM, and model 4 with constraint

$$\underbrace{\Pr(\text{English word} \mid \text{AMR concept})}_{\text{IBM model with source = AMR}} \propto \underbrace{\Pr(\text{AMR concept} \mid \text{English word})}_{\text{IBM model with target = AMR}}$$

and similar constraint for the distortion tables.

EM Alignment

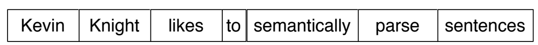
- (Picture of AMR graph, linearization+preprocessing, and sentence)
- Say run IBM model 2, and get alignments (show)
- Talk about constraint?
- Project alignments to AMR graph

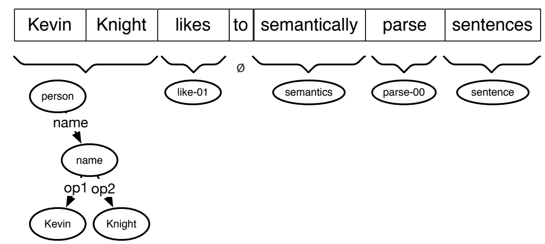
Structured Prediction

Input

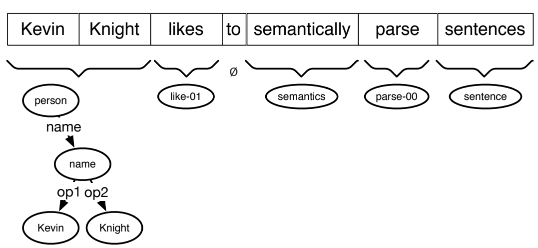
Output

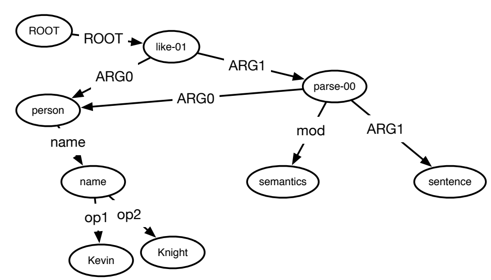
Concept ID





Relation ID





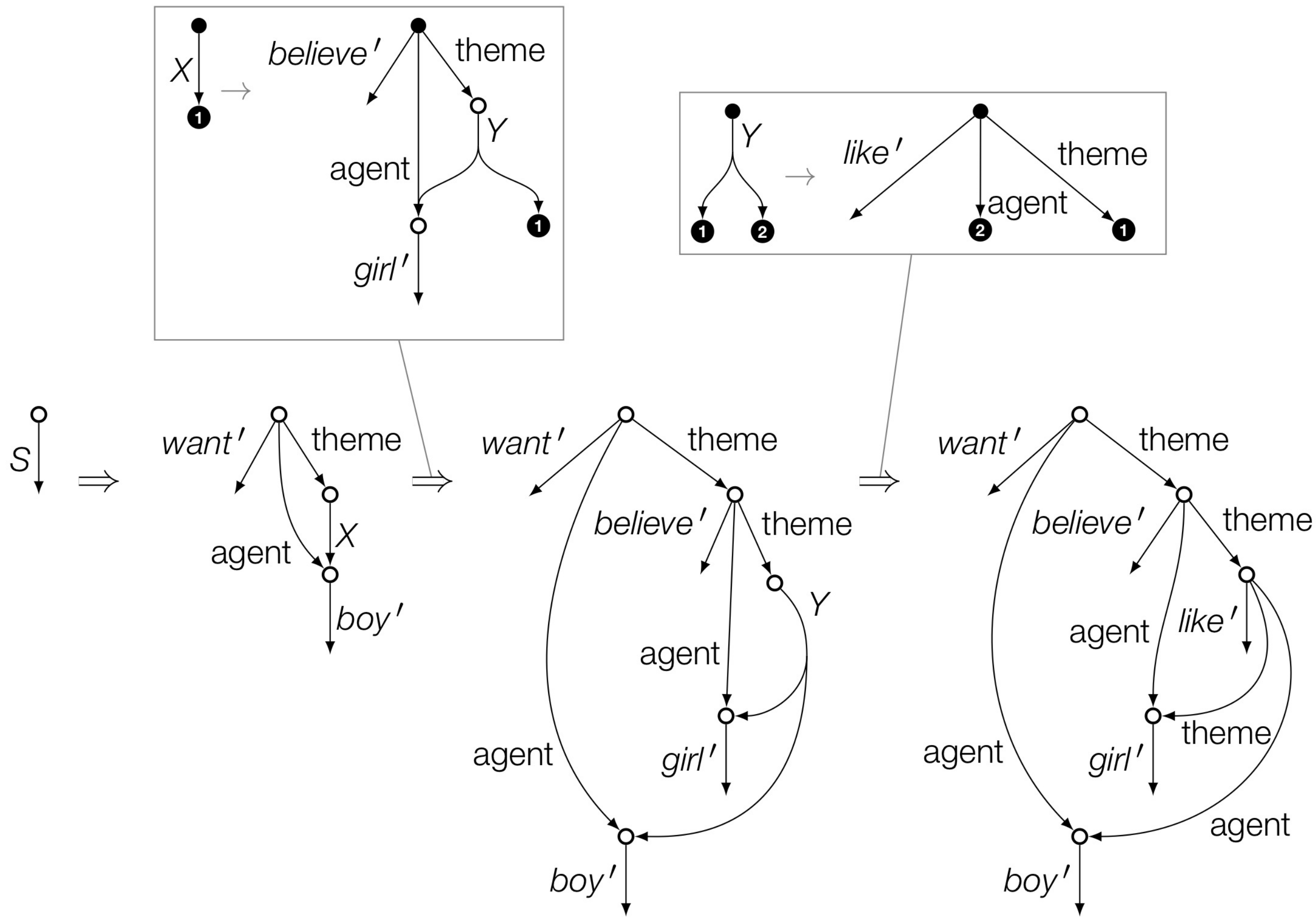
Decoder

- (recurrence relation)

Features

- Bias
- NER
- Length
- Fragment given words

Hyperedge replacement grammar



Synchronous CFGs

- (Example, English & French)

Synchronous Grammars

- S -> NP1 VP1, VP1 NP2 (CFG on both sides)
- S -> NP1 VP2 NP3, NP3 VP2 NP1 (CFG on one side, TSG on other)
- S -> (cfg on one side, HRG on other)
- S -> (cfg on one side, DAG automata on other)

Synchronous graph grammars can be used to build a string to graph parser

How to use them

- Parsing (string to graph)
  - Synchronous with source-side CFG, string to graph transducer
- Generation (graph to string)
  - Synchronous with target-side CFG, graph to string transducer
- Transduction (graph to graph, or tree to graph or graph to tree)
  - Synchronous grammar or transducer

	Strings		Graphs	
	FSA	CFG	DAG automata	HRG
reentrancies	–		yes	local
transducers	yes	yes	to tree	**yes**
probabilistic	yes	yes	**yes**	yes
intersection with finite-state	yes	yes	yes	**yes**
summation	yes	yes	**yes**	yes
recognition	$O(n)$	$O(n^3)$	$\boldsymbol{O(q^{k+1}n)}$	$\boldsymbol{O(n^{k+1})}$ connected, bounded degree
implemented	yes	yes	**yes**	**yes**

- Summary of everything so far
- Grammar-based, vs transition and graph-based
- Similar to syntactic parsing

Results

	ACL 2014	**Retrained**
Concept Identification	76% F_1	73% F_1
Full System (gold concepts)	80% Smatch	81% Smatch
Full System	58% Smatch	62% Smatch