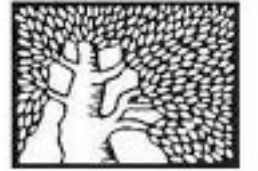




UPPSALA
UNIVERSITET



מכון
ויצמן
למדע

Parsing

Morphologically Rich Languages (PMRL)

**Dependency Parsing
for English/MRLs**

Reut Tsarfaty

reut.tsarfaty@weizmann.ac.il

Yesterday@PMRL

- ✓ ● Day 1: Introduction
- ✓ ● Day 2: Phrase-structure
- ✓ ● Inference for English/MRLs

➔ Learning for English/MRLs

- Day 3: Dependency-structure
- Day 4: Relational-Realizational
- Day 5: Evaluation and Multilinguality

Yesterday@PMRL

- Day 1: Introduction
- Day 2: Phrase-structure
- Inference for English/MRLs
- Learning for English/MRLs
- ➔ Day 3: Dependency-structure
- Day 4: Relational-Realizational
- Day 5: Evaluation and Multilinguality

Today@PMRL

✓ ● Day 1: Introduction

✓ ● Day 2: Phrase-structure

➔ ● Day 3: Dependency-structure

- Inference for English/MRLs

- Learning for English/MRLs

- Day 4: Relational-Realizational

- Day 5: Evaluation and Multilinguality

Motivation

"Dependency-based methods for syntactic parsing have become increasingly popular in natural language processing in recent years. One of the reasons for their success is that they have been shown to work reliably for a wide range of typologically different languages" (Kuebler et al 2009)

<http://www.amazon.de/Dependency-Synthesis-Lectures-Language-Technologies/dp/1598295969>

However...

"It is interesting to see that the classes are more easily definable via language characteristics than via characteristics of the data sets. The split goes across training set size, original data format [...], sentence length, percentage of unknown words, number of dependency labels, and ratio of (C)POSTAGS and dependency labels. The class with the highest top scores contains languages with a rather impoverished morphology." (CoNLL shared task 2007)

<http://acl.ldc.upenn.edu/D/D07/D07-1096.pdf>

Shared Task 2007



Shared Task 2007

Low (76.31–76.94):

- Arabic, Basque, Greek

- Medium (79.19–80.21):

- Czech, Hungarian, Turkish

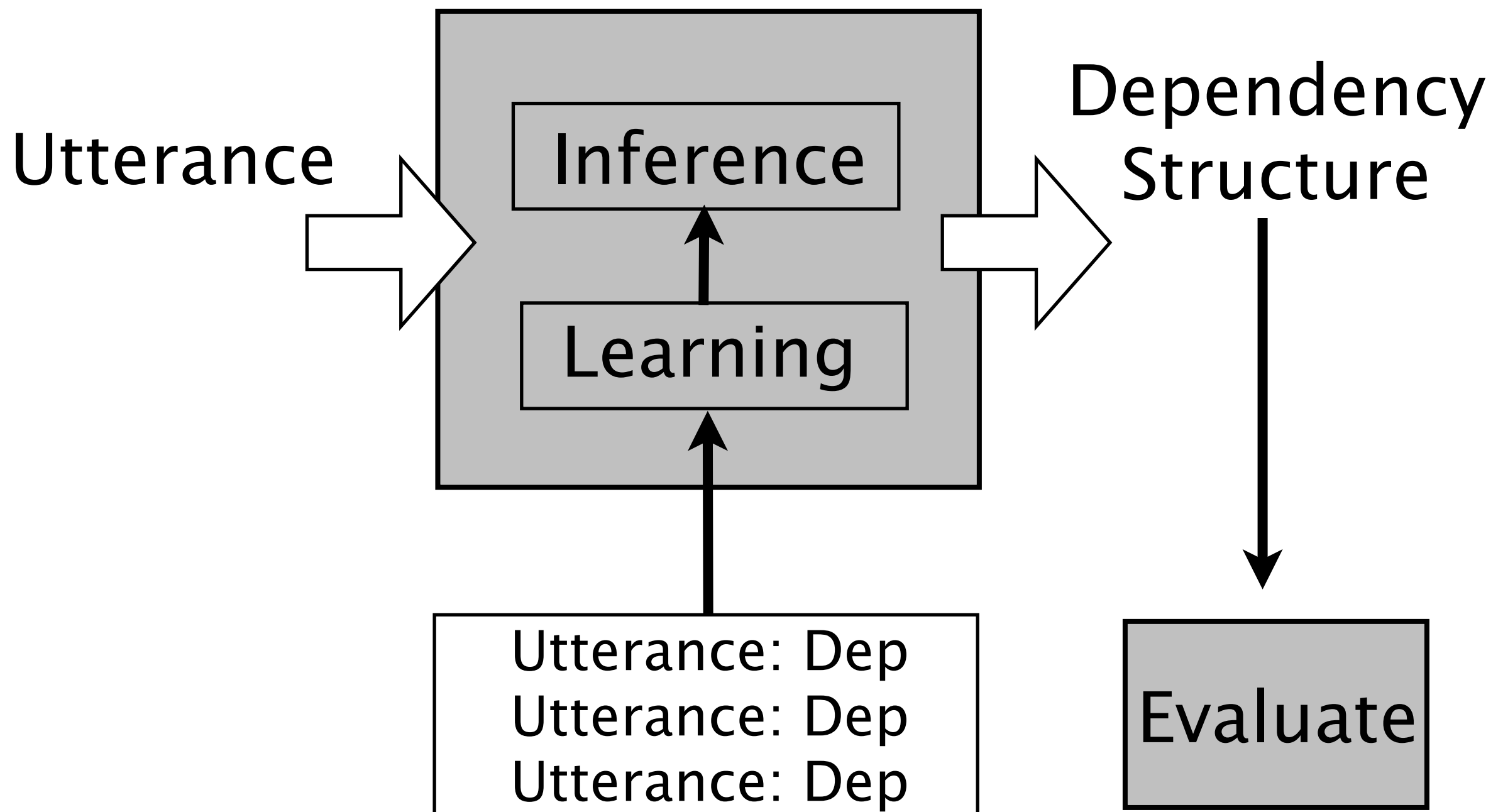
- High (84.40–89.61):

- Catalan, Chinese, English,



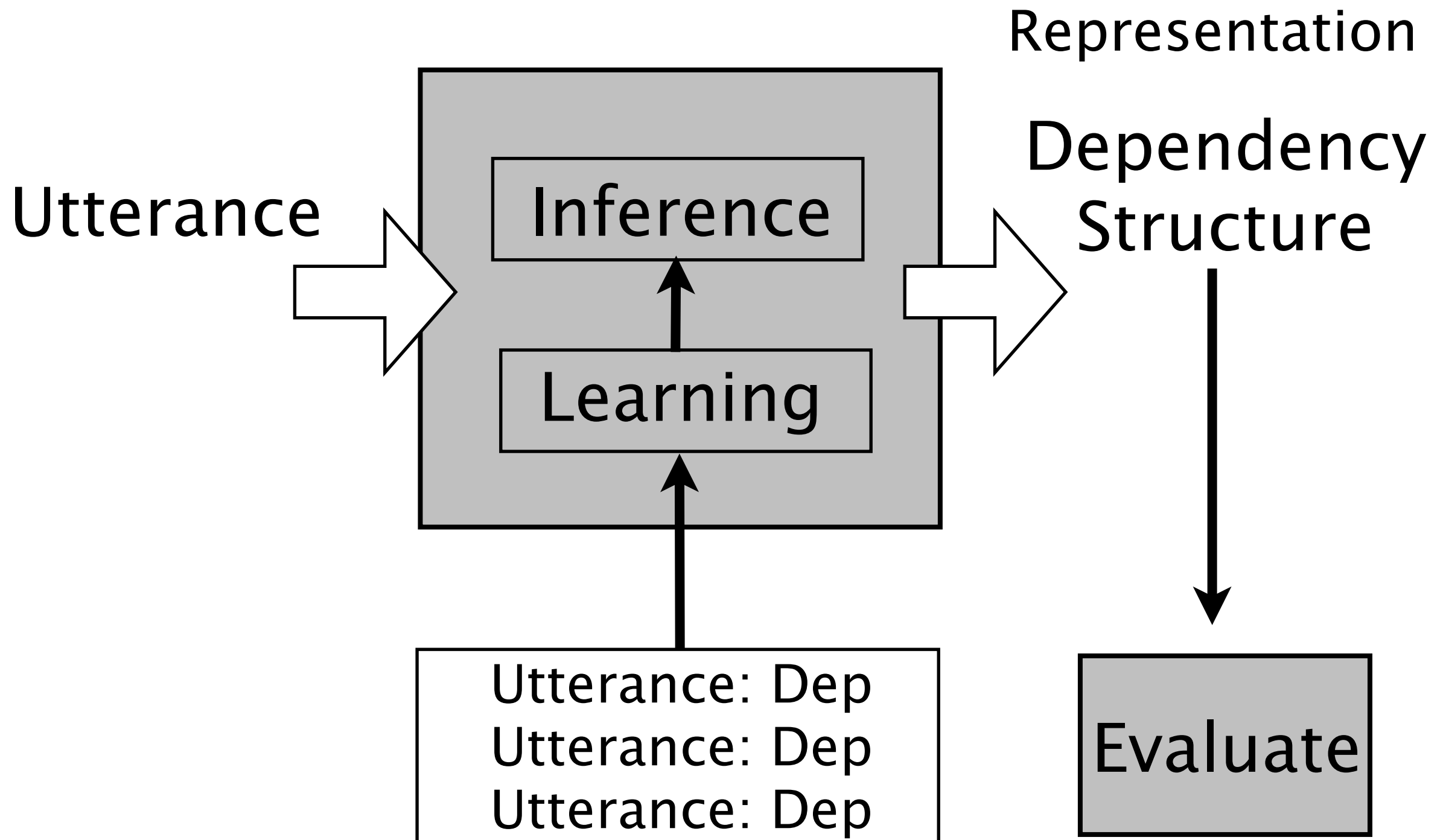
Today@PMRL

DS for MRLs



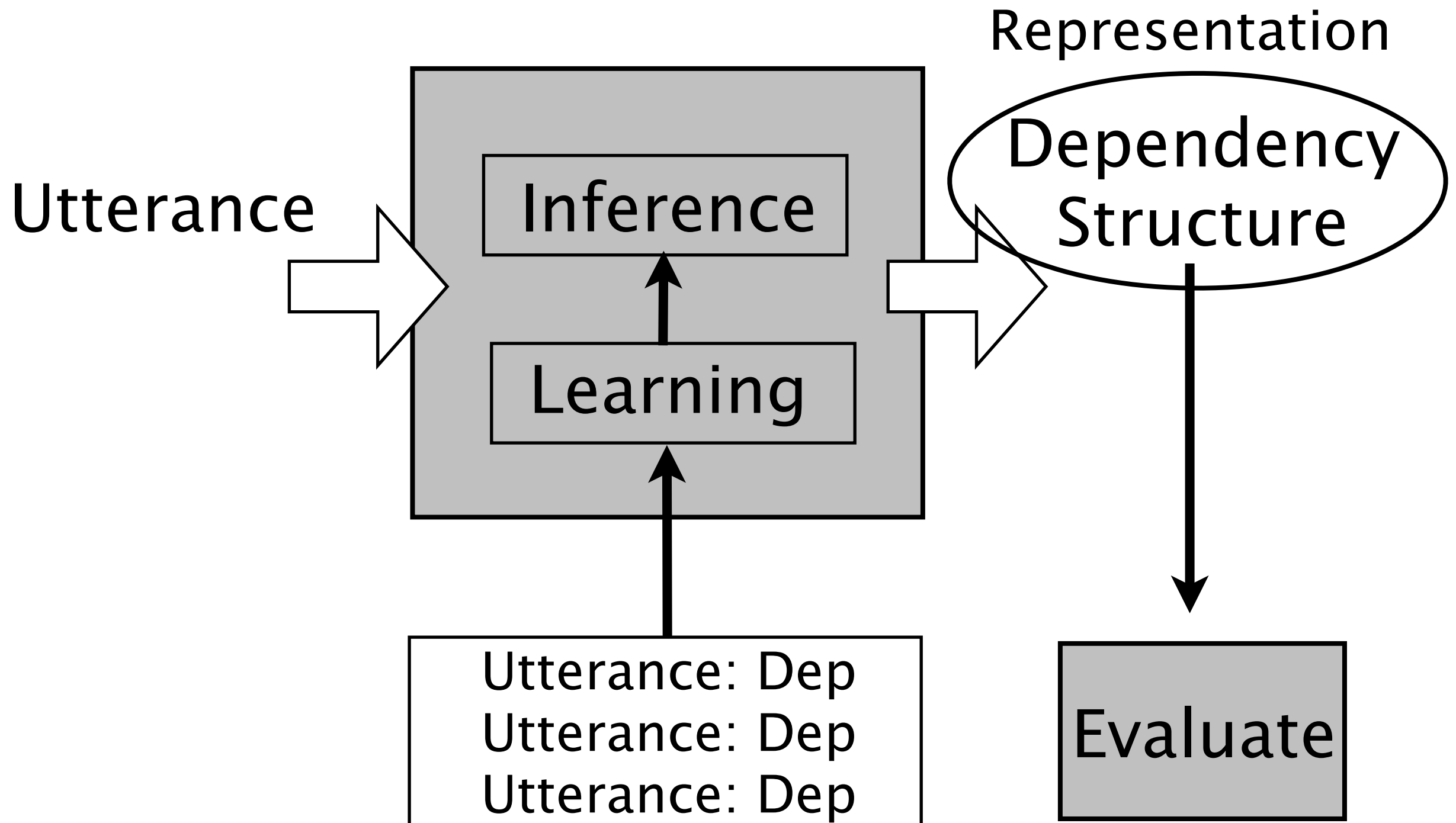
Today@PMRL

DS for MRLs



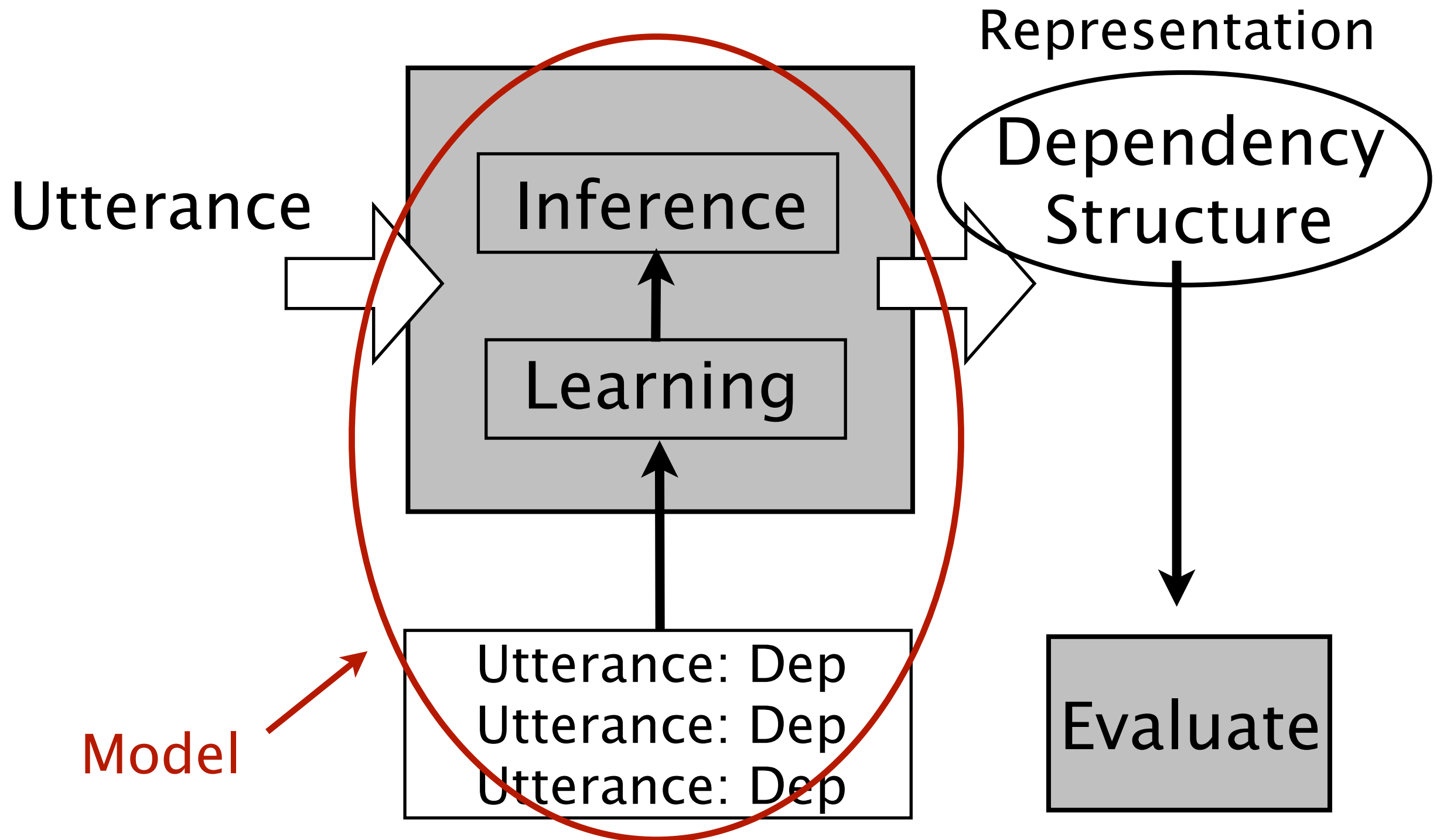
Today@PMRL

DS for MRLs



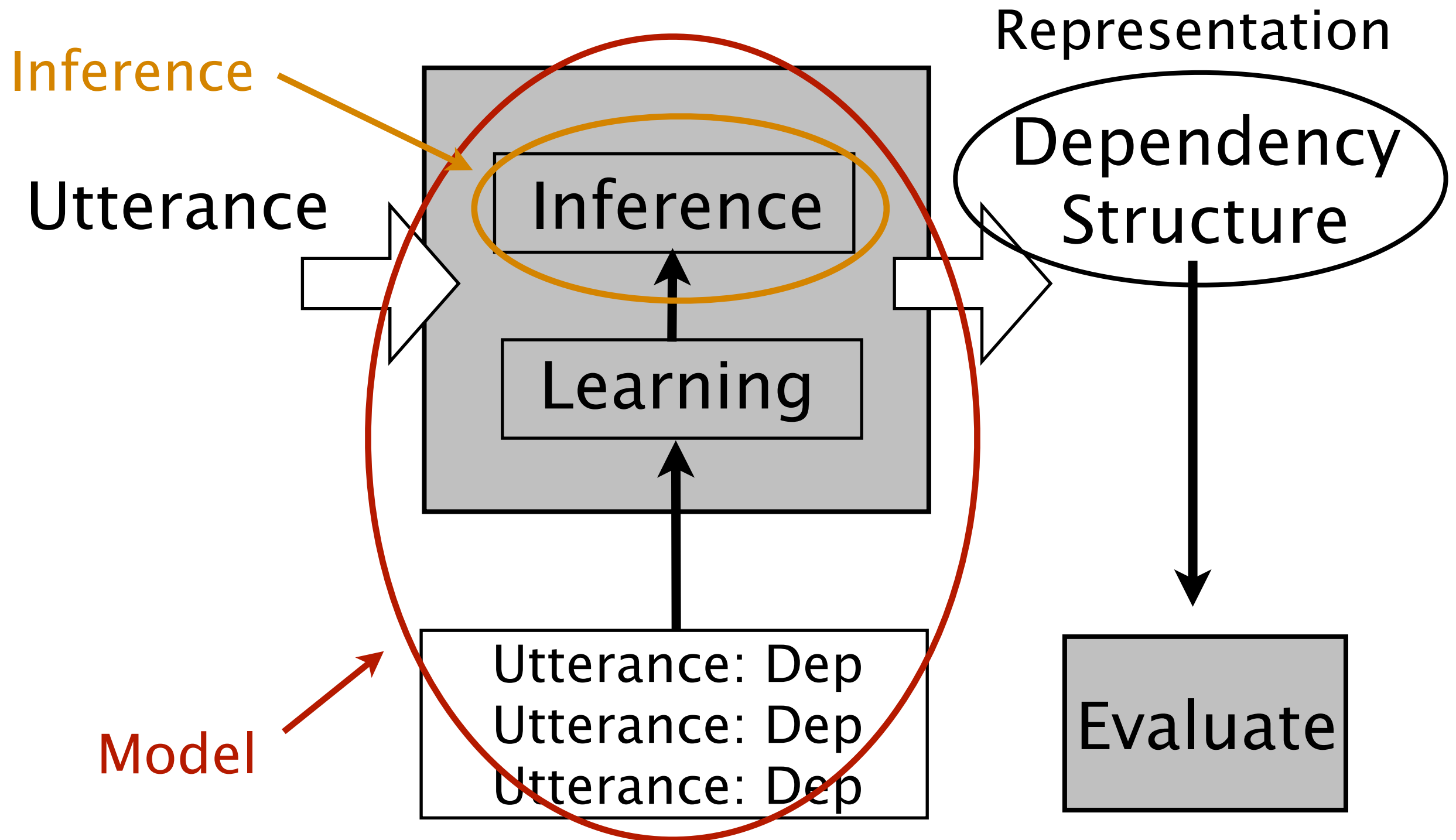
Today@PMRL

DS for MRLs



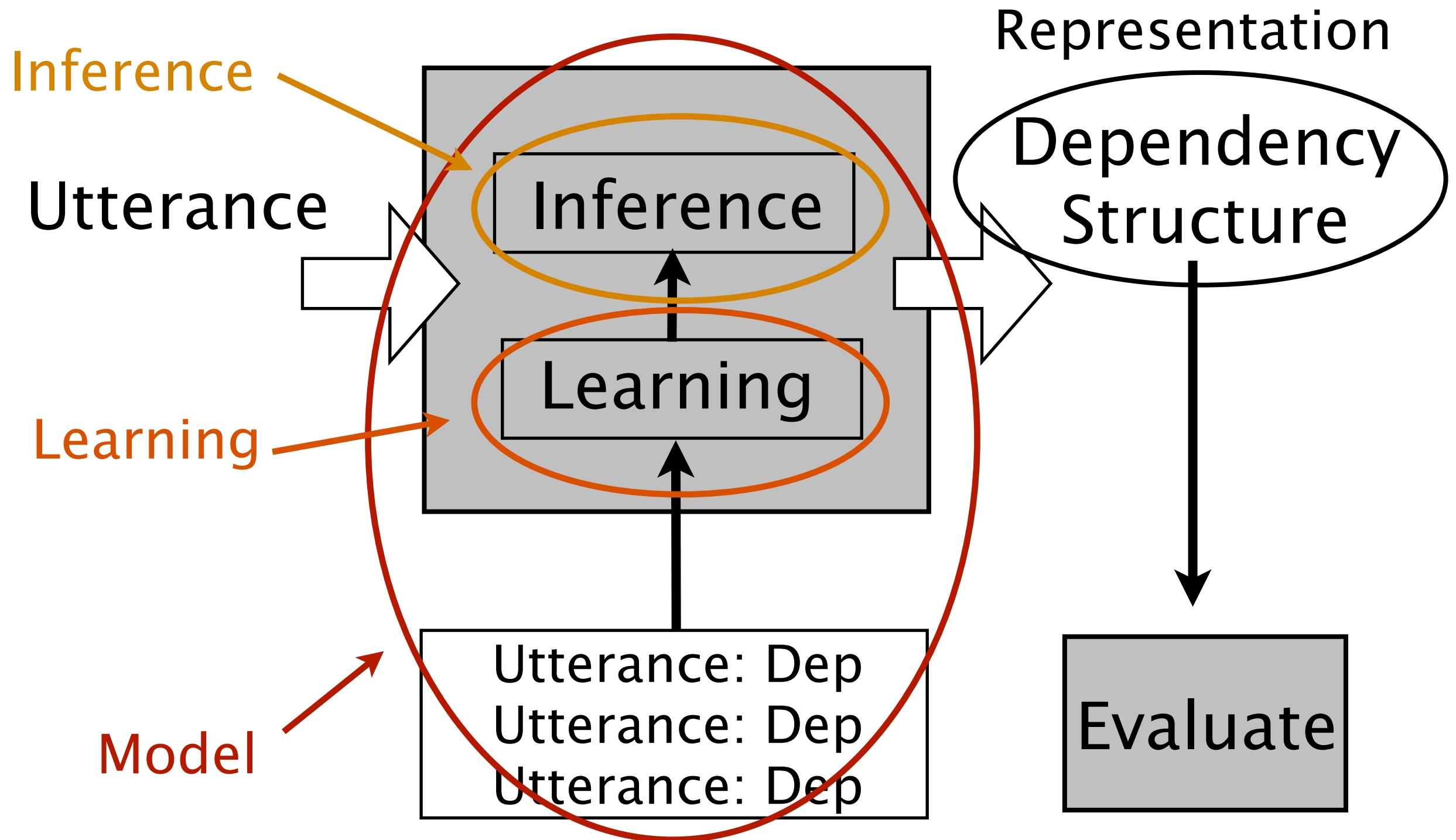
Today@PMRL

DS for MRLs



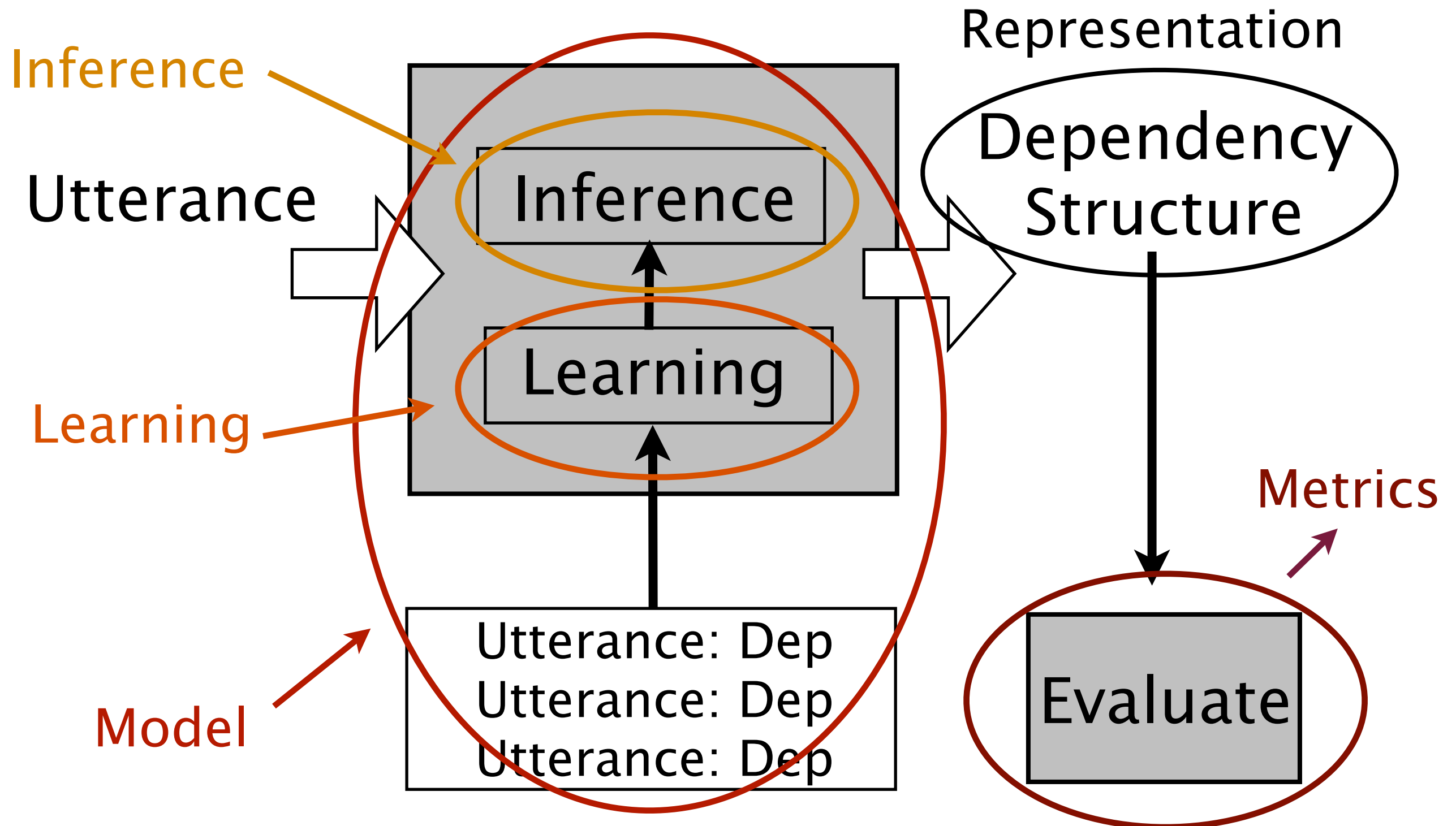
Today@PMRL

DS for MRLs



Today@PMRL

DS for MRLs



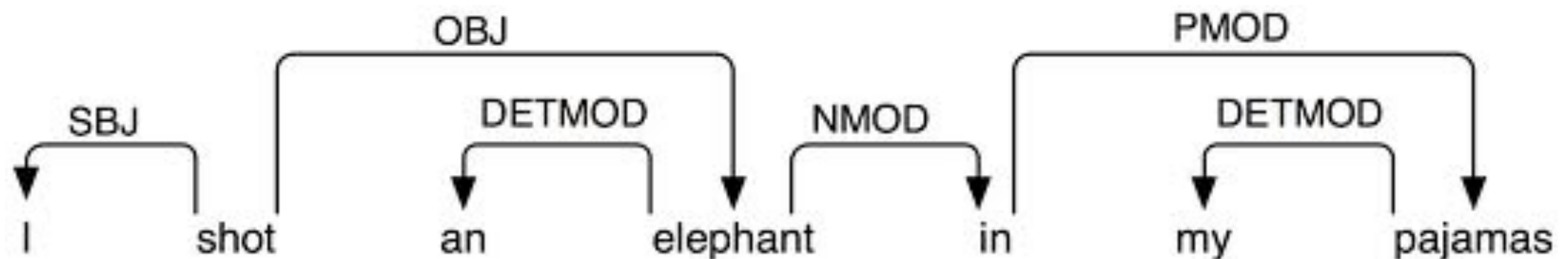
Introducing English Dependency- based Parsing

Architectural Decisions

- Representation: Dependency Trees
- Model: ?
- Inference: ?
- Learning: ?
- Evaluation: ?

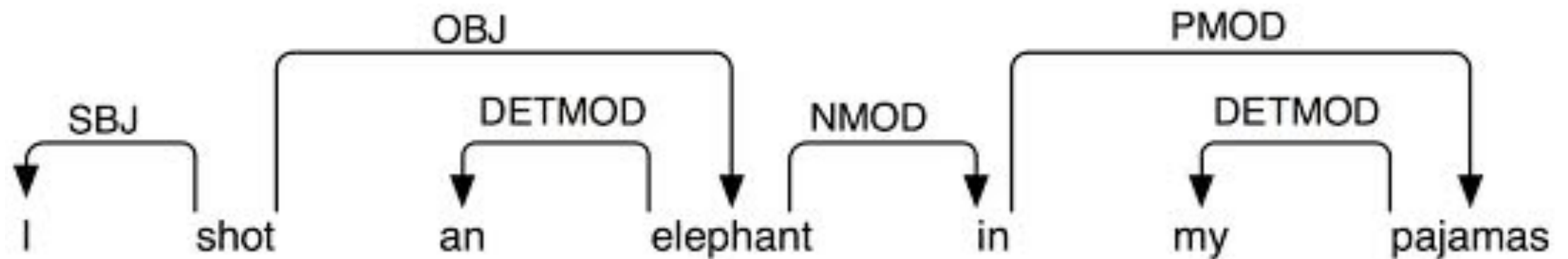
Representation

- Assume:
 - A finite vocabulary W
 - An artificial root w_0
 - A finite set of relation labels R



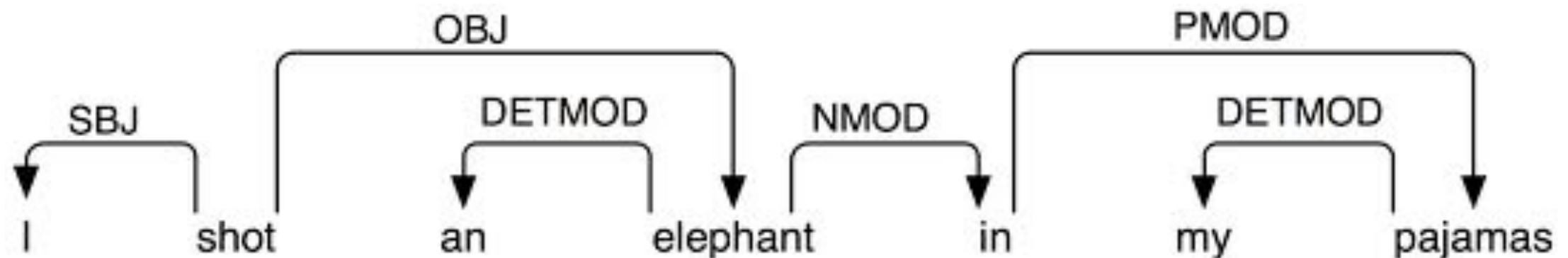
Representation

- A dependency graph $G(V, A)$ is a labeled dependency graph for $S = w_1 \dots w_n$ s.t.
 - Nodes: $V = w_0 \cup \{w_1 \dots w_n\}$
 - Arcs: $A \subset V \times R \times V$
 - $(w_i, r_1, w_j) \in A \wedge (w_i, r_2, w_j) \in A \rightarrow r_1 = r_2$



Representation

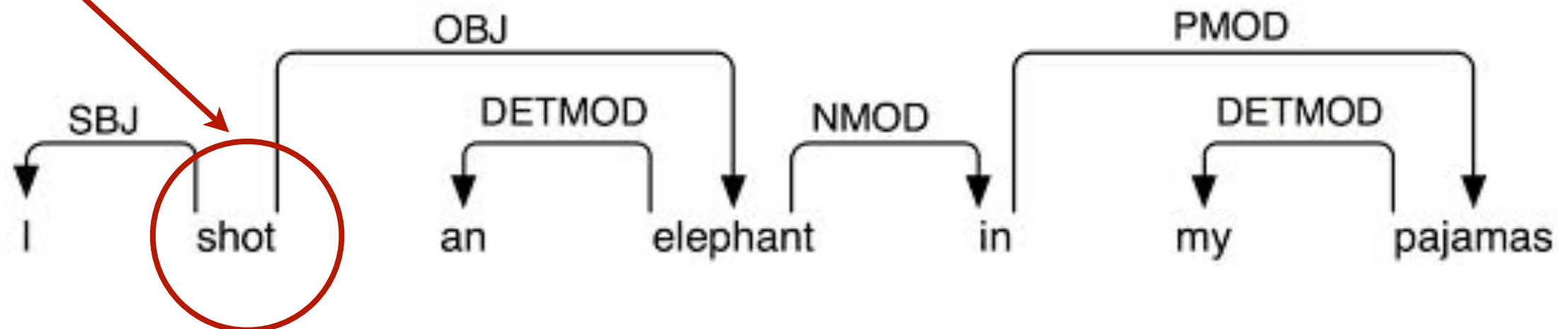
- A dependency graph $G(V, A)$ is a dependency tree for $S = w_1 \dots w_n$ iff
 - It is directed
 - It is acyclic
 - it obeys the single head property
 - it obeys the single root property



Representation

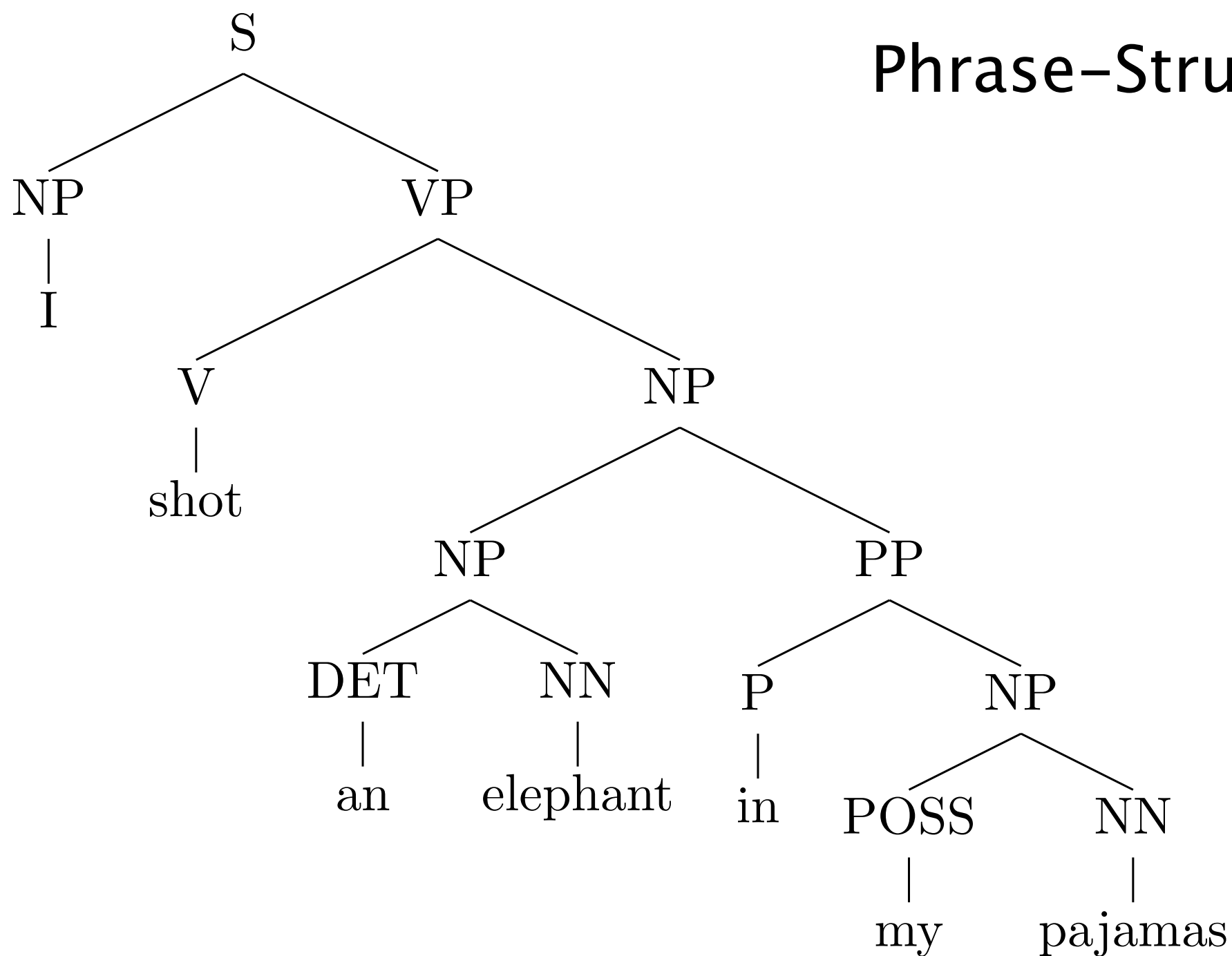
- A dependency graph $G(V, A)$ is a dependency tree for $S = w_1 \dots w_n$ iff
 - It is directed
 - It is acyclic
 - it obeys the single head property
 - it obeys the single root property

Root



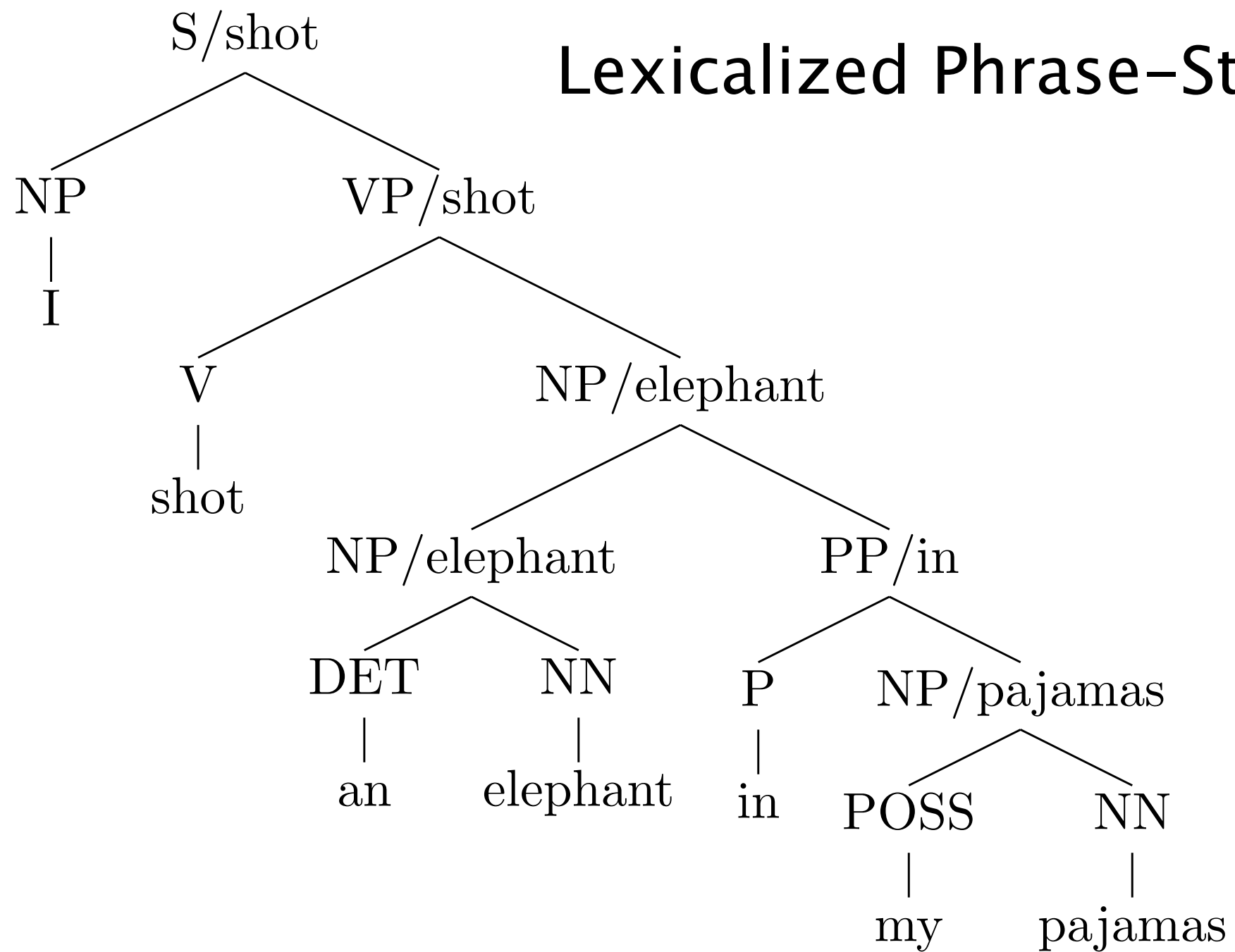
Representation

Phrase-Structures



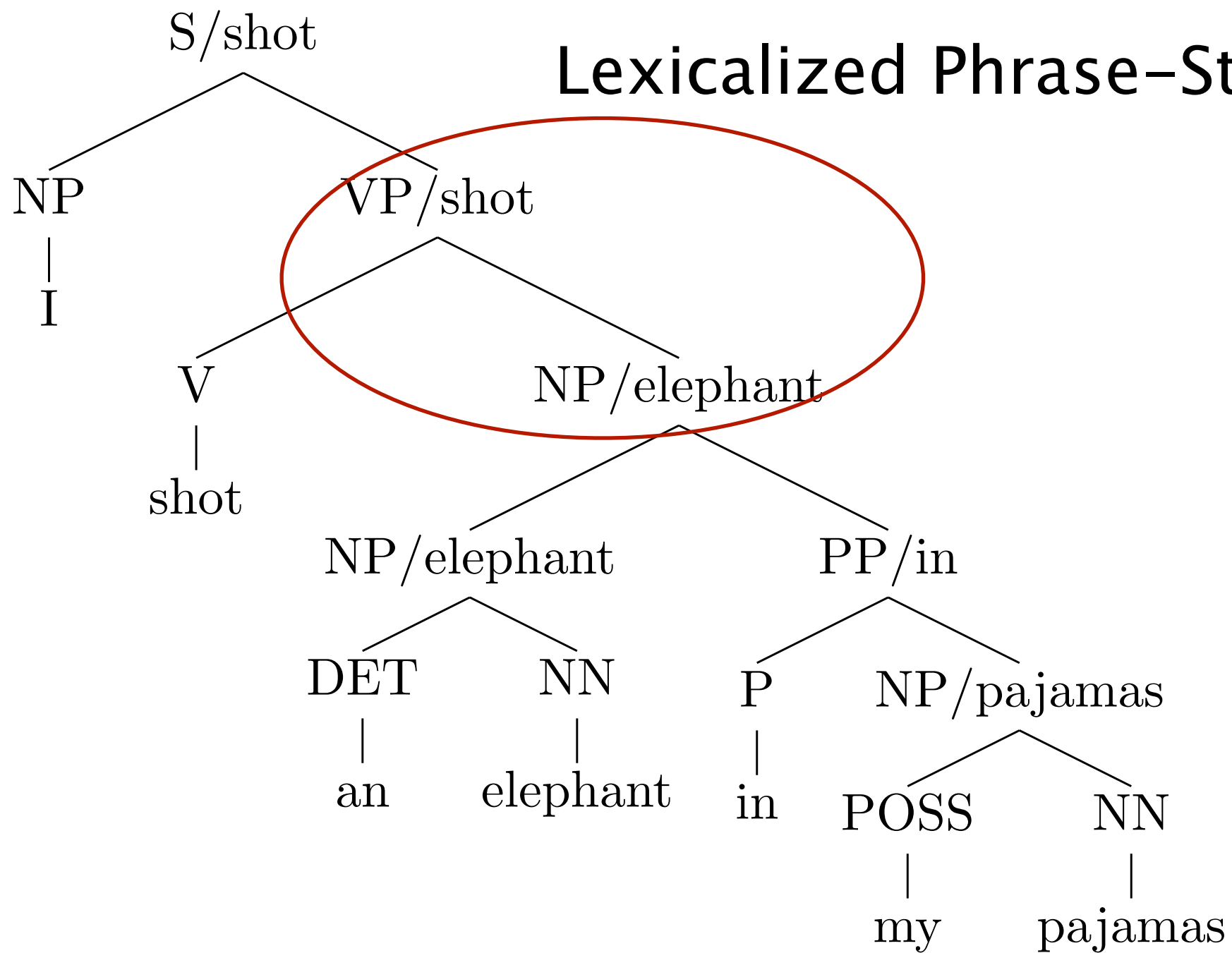
Representation

Lexicalized Phrase-Structures



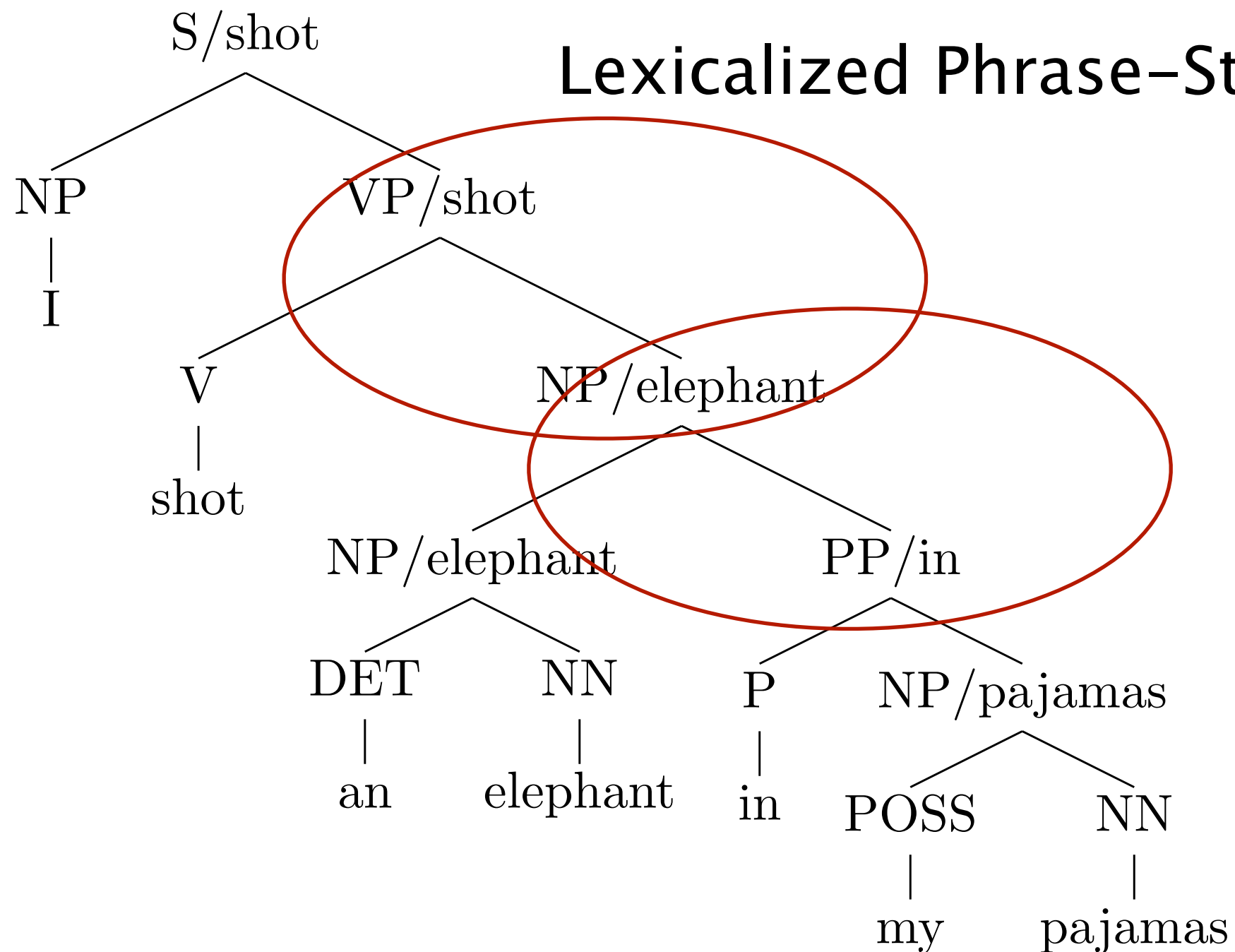
Representation

Lexicalized Phrase-Structures



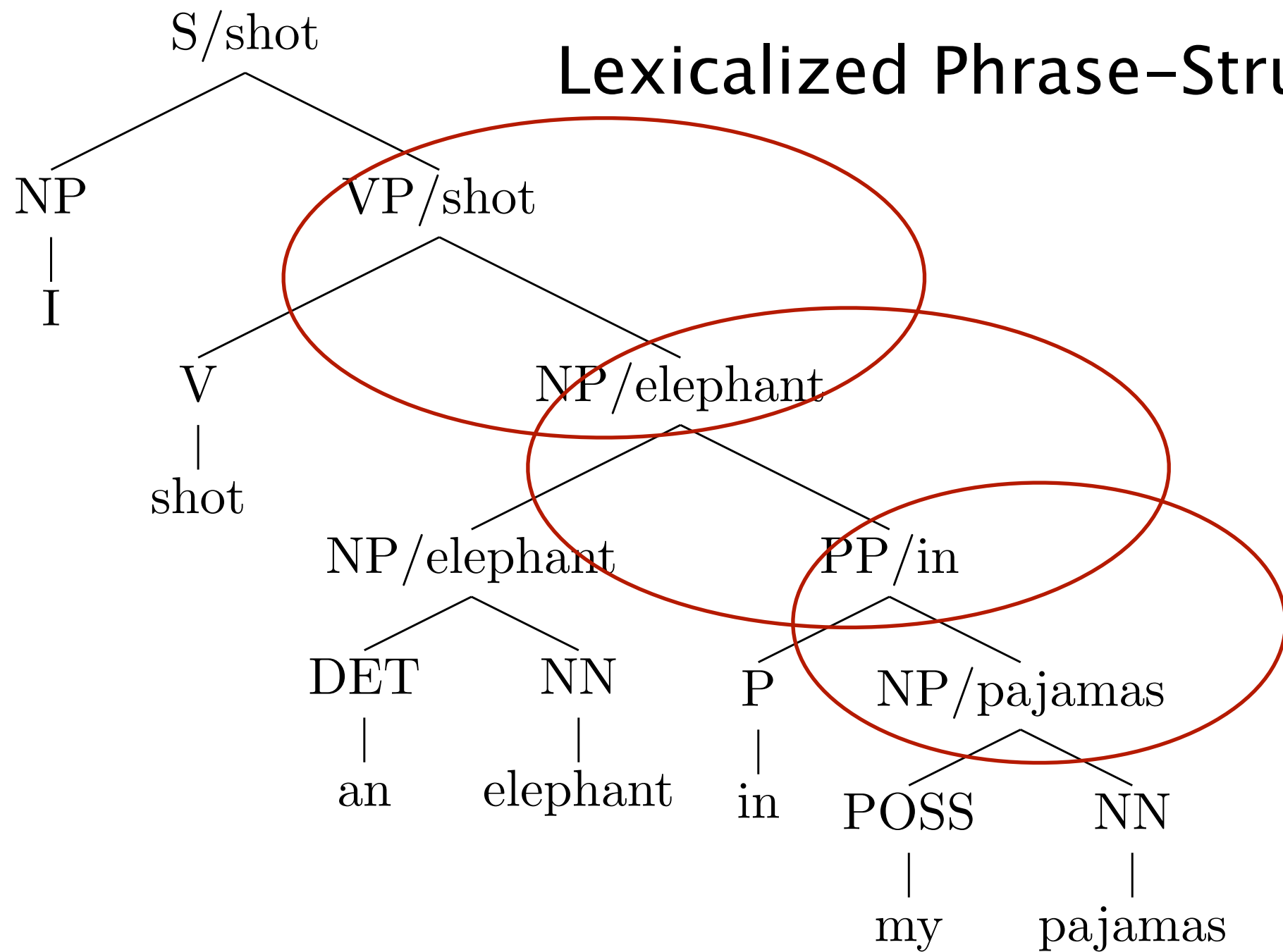
Representation

Lexicalized Phrase-Structures



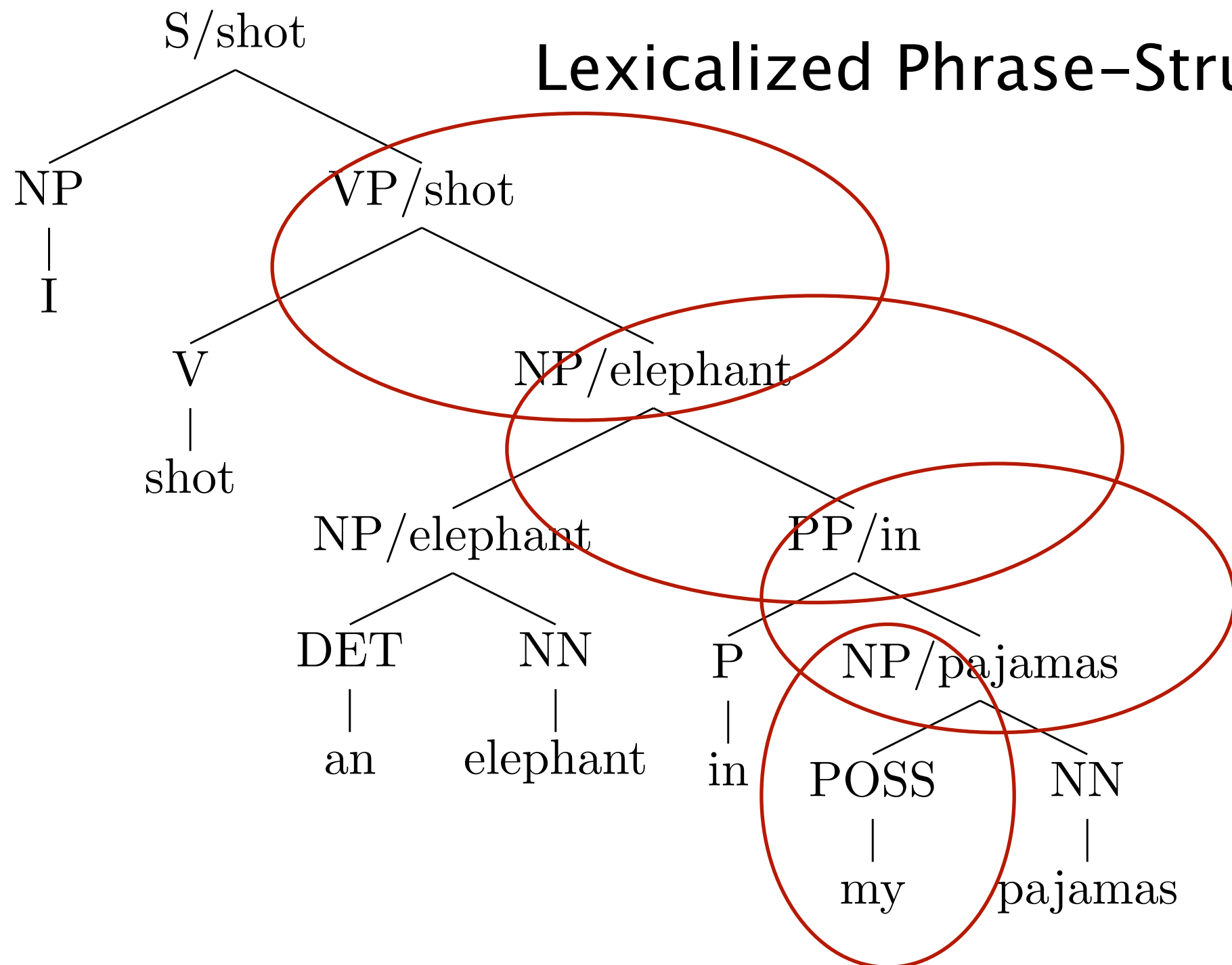
Representation

Lexicalized Phrase-Structures



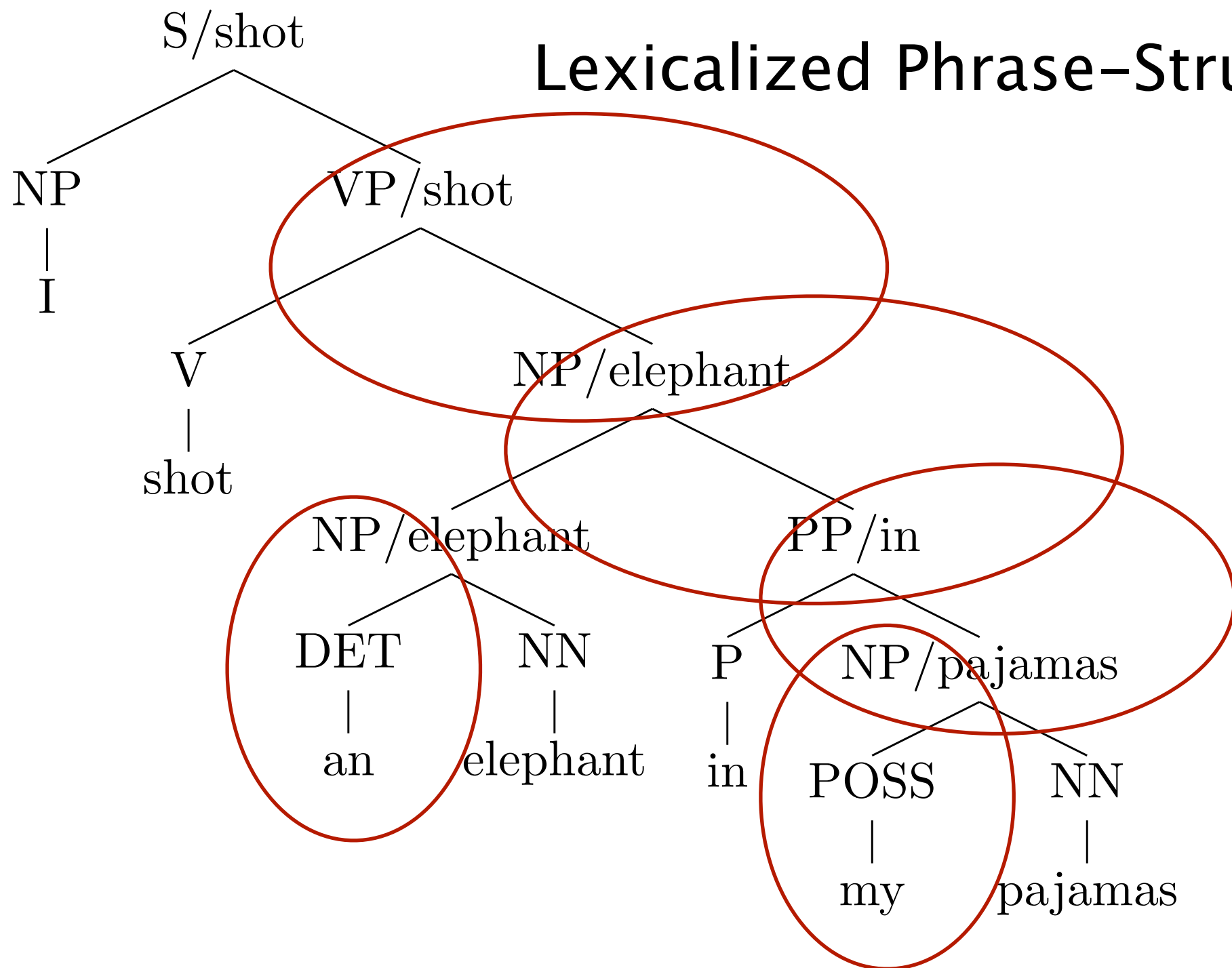
Representation

Lexicalized Phrase-Structures



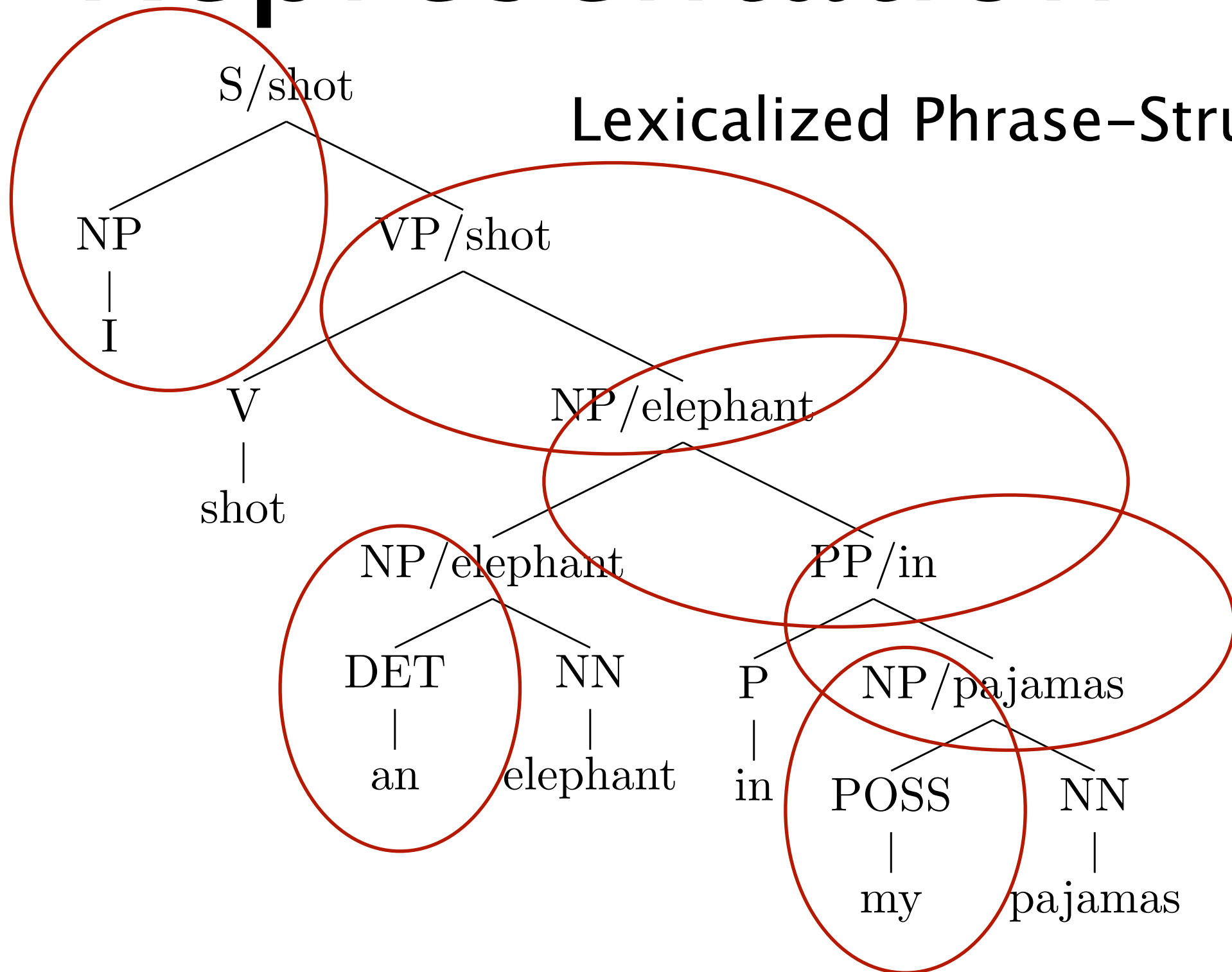
Representation

Lexicalized Phrase-Structures



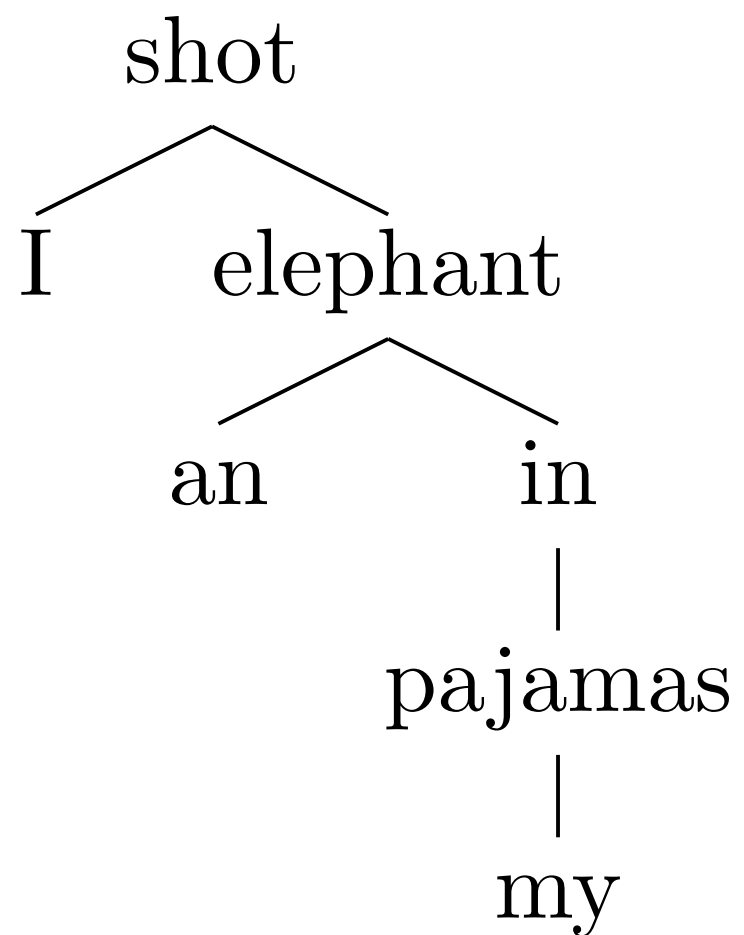
Representation

Lexicalized Phrase-Structures



Representation

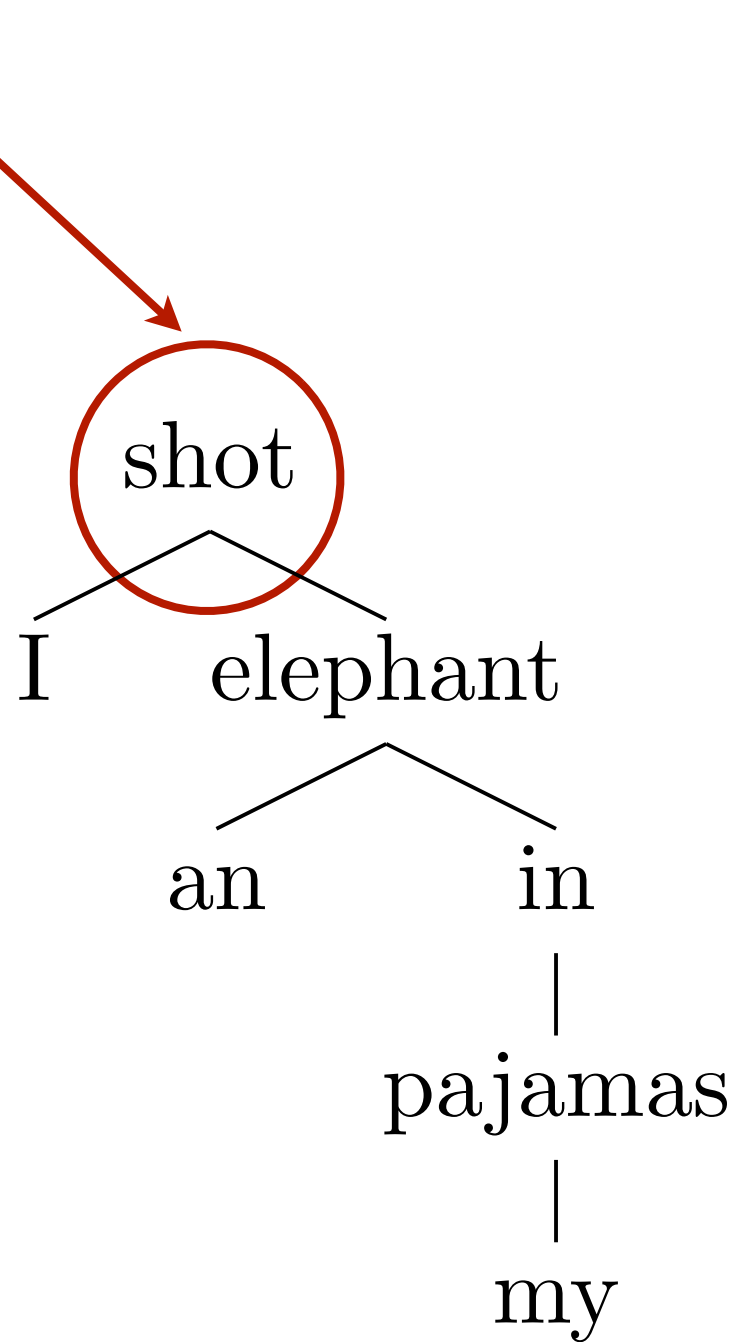
Bi-Lexical Dependency



Representation

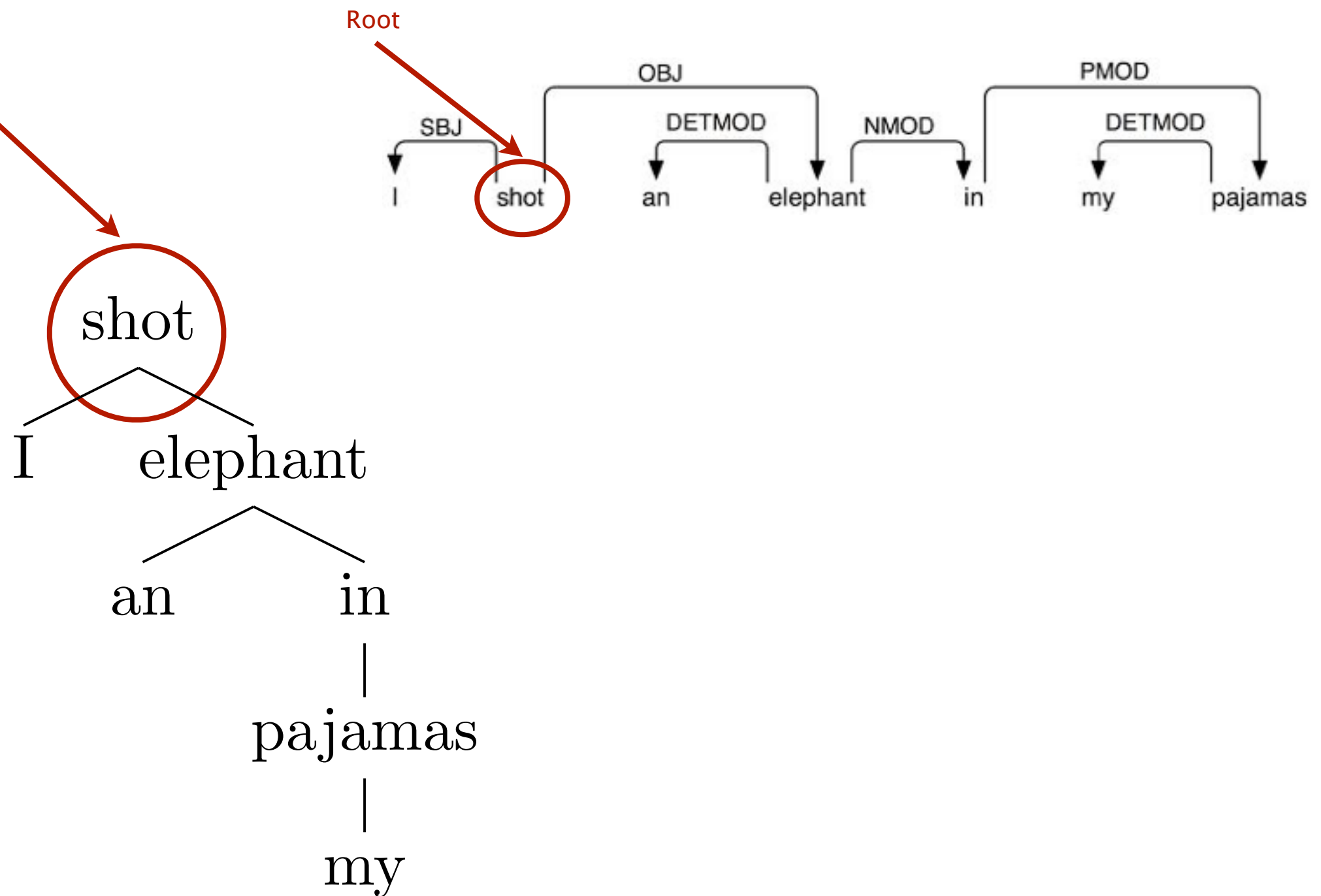
Bi-Lexical Dependency

Root



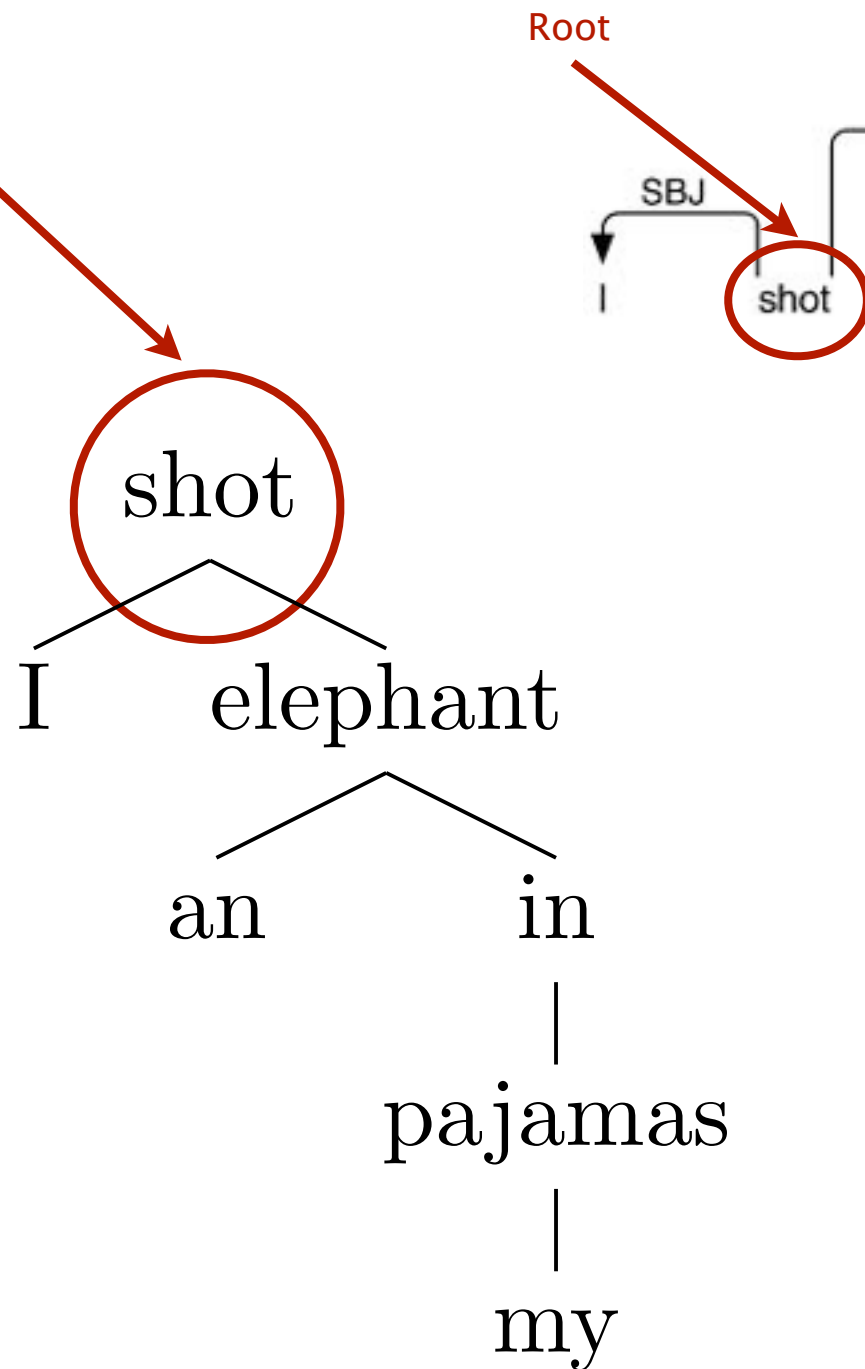
Representation

Root

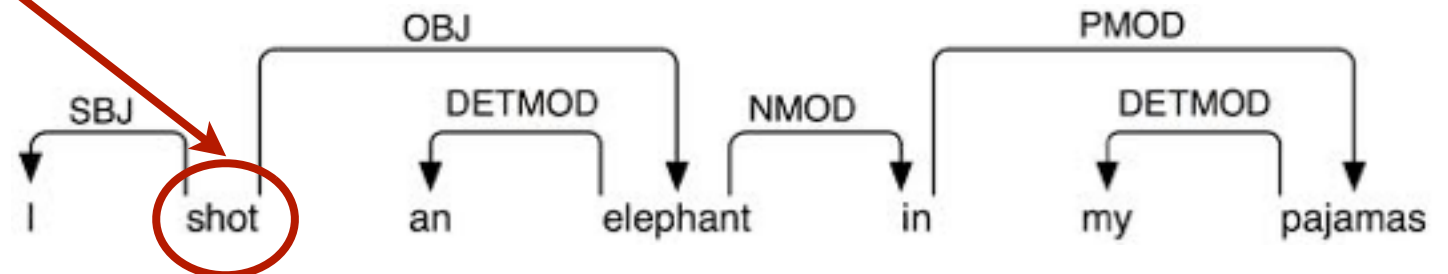


Representation

Root

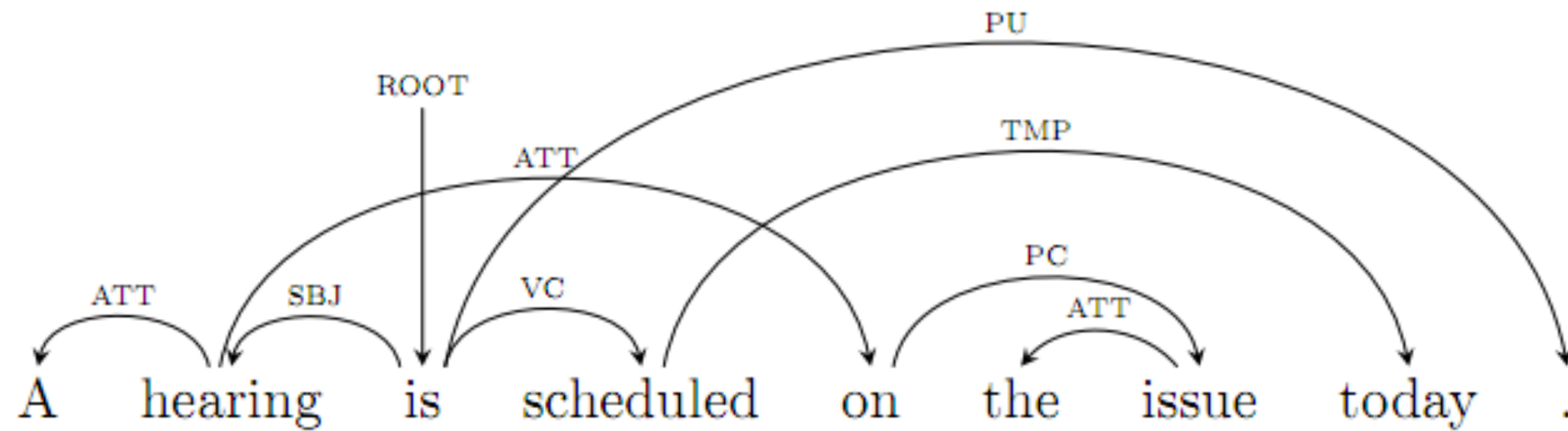


Root

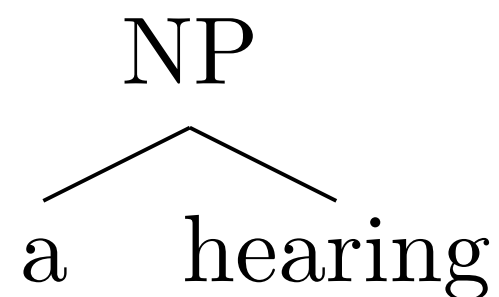
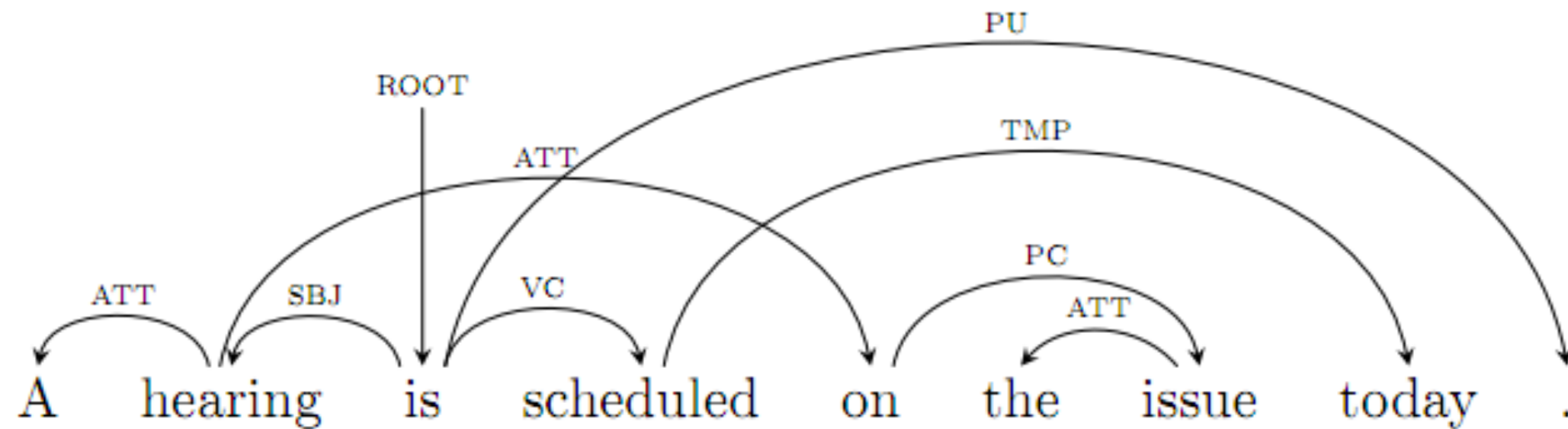


These dependency
representations
are equivalent

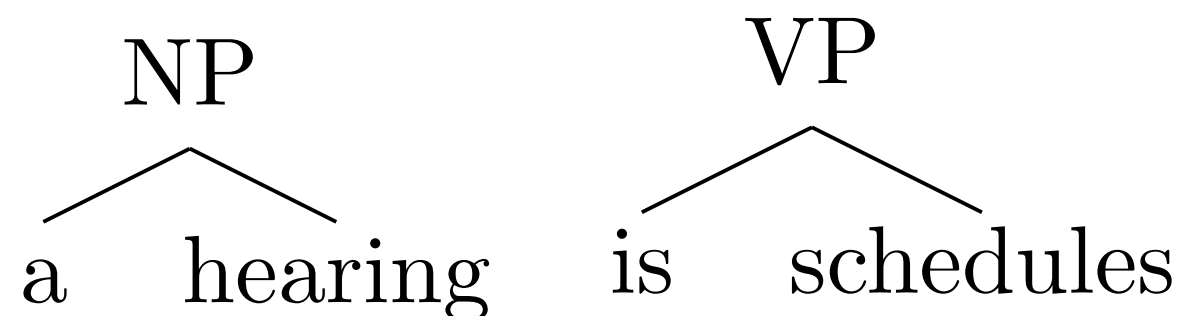
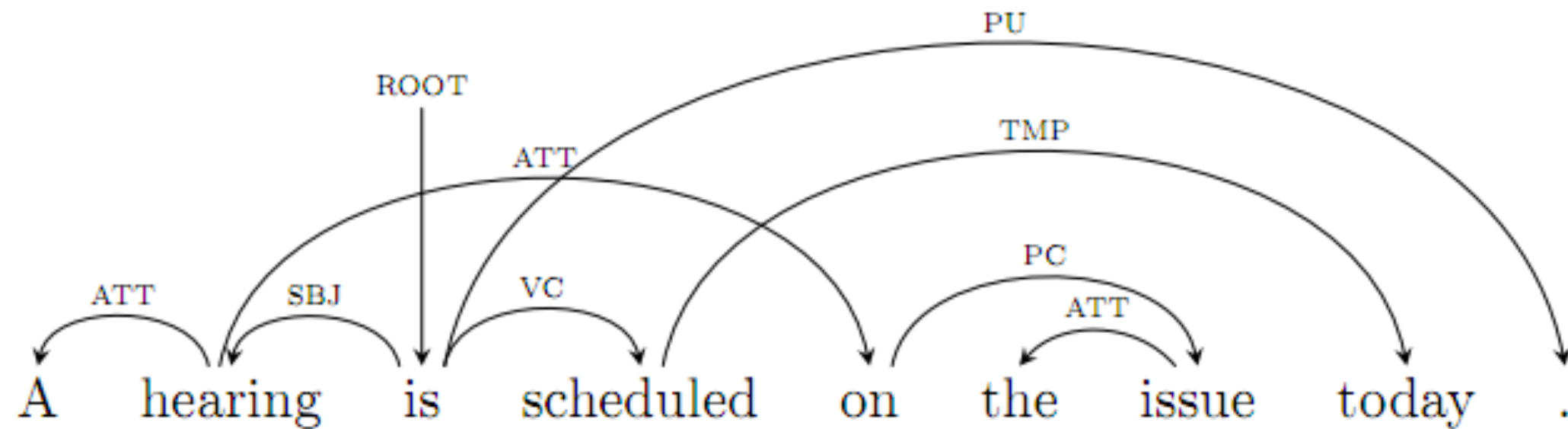
A Comment on Nonprojectivity



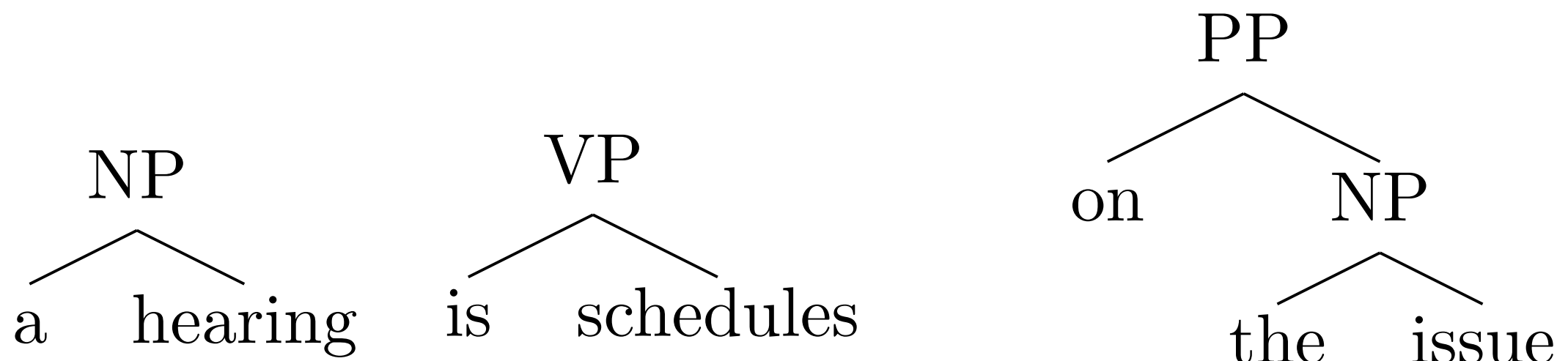
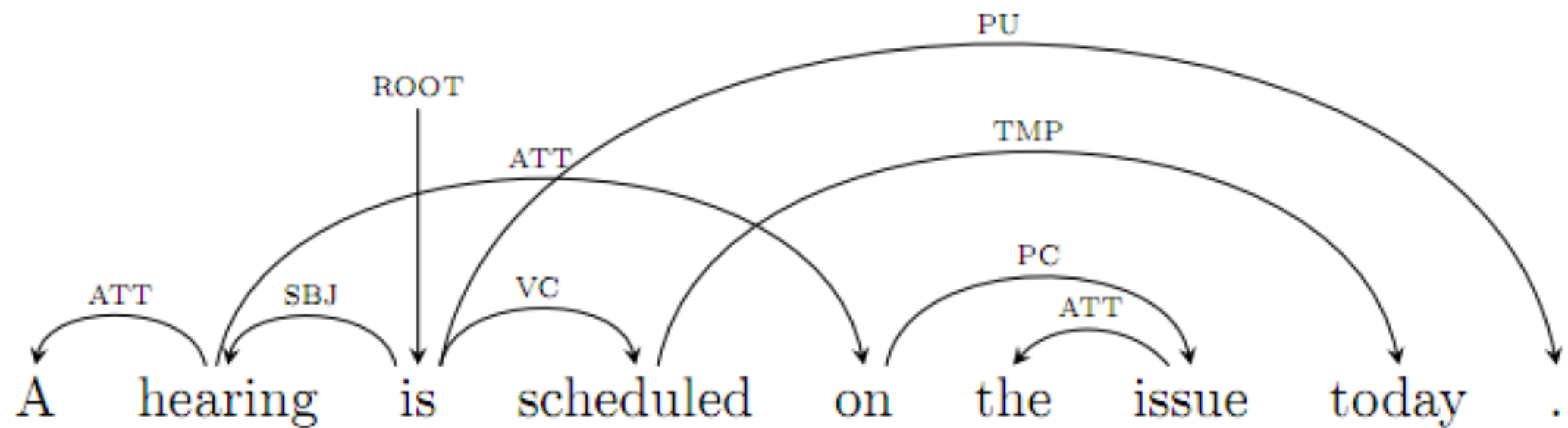
A Comment on Nonprojectivity



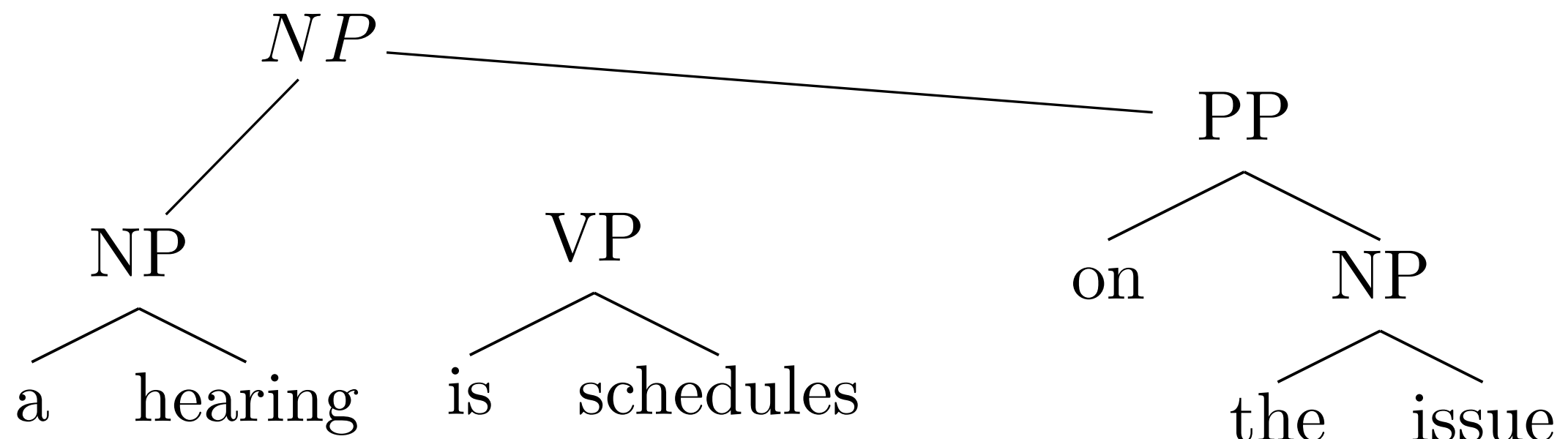
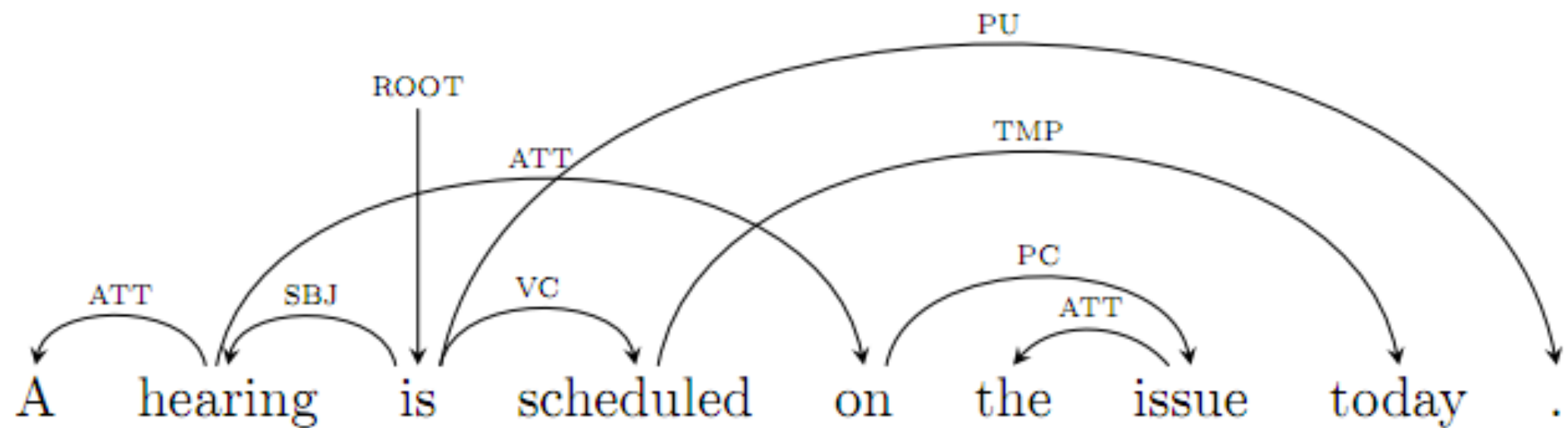
A Comment on Nonprojectivity



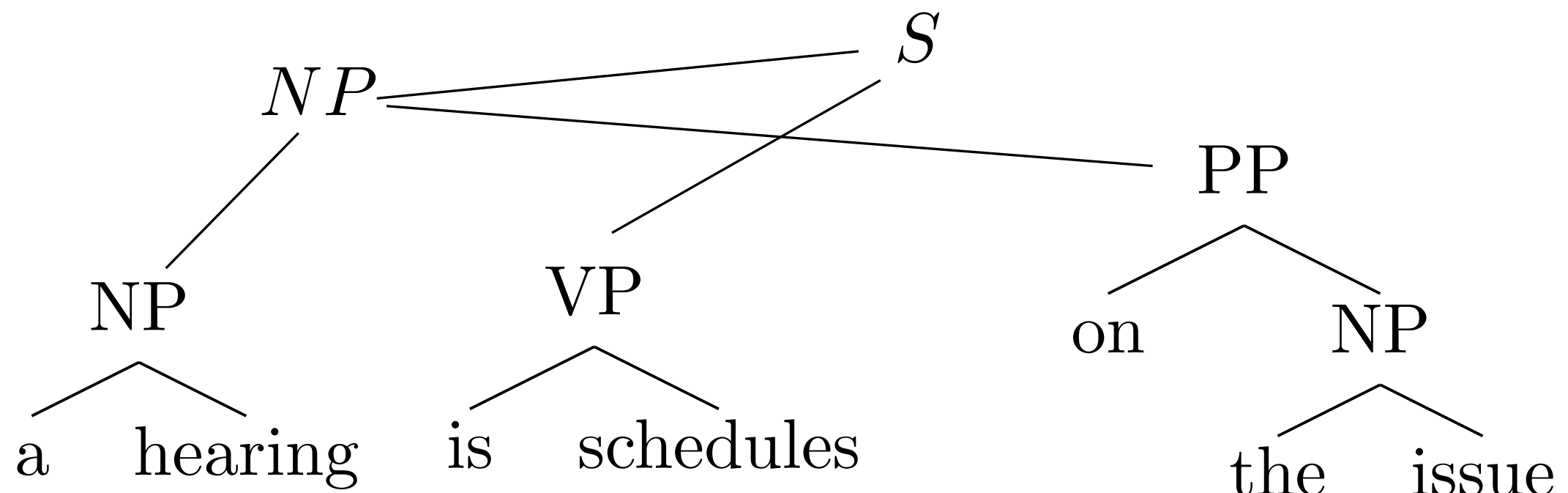
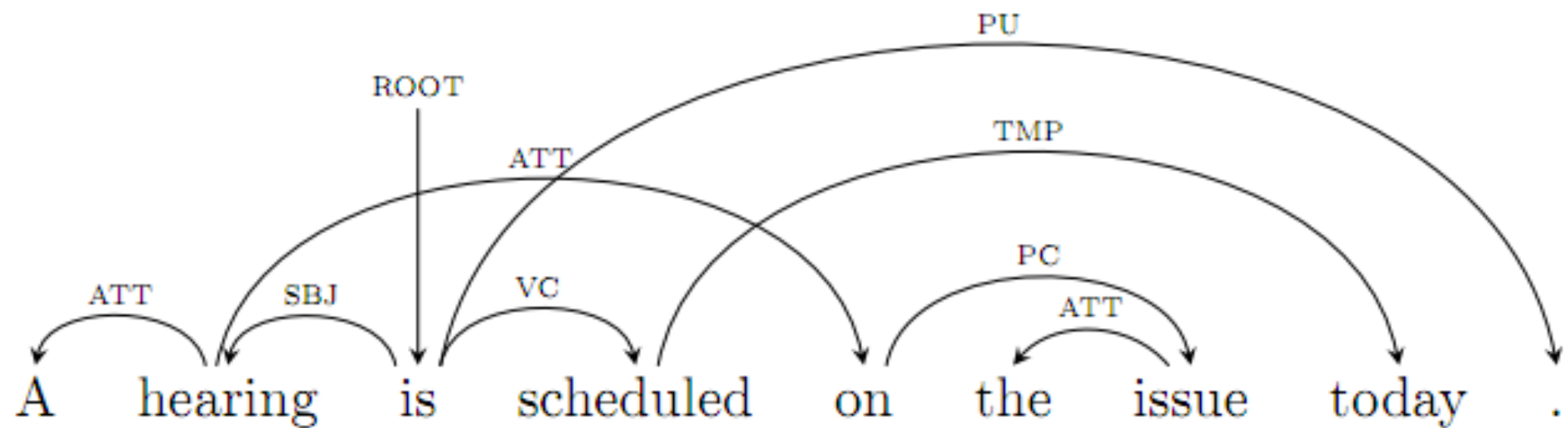
A Comment on Nonprojectivity



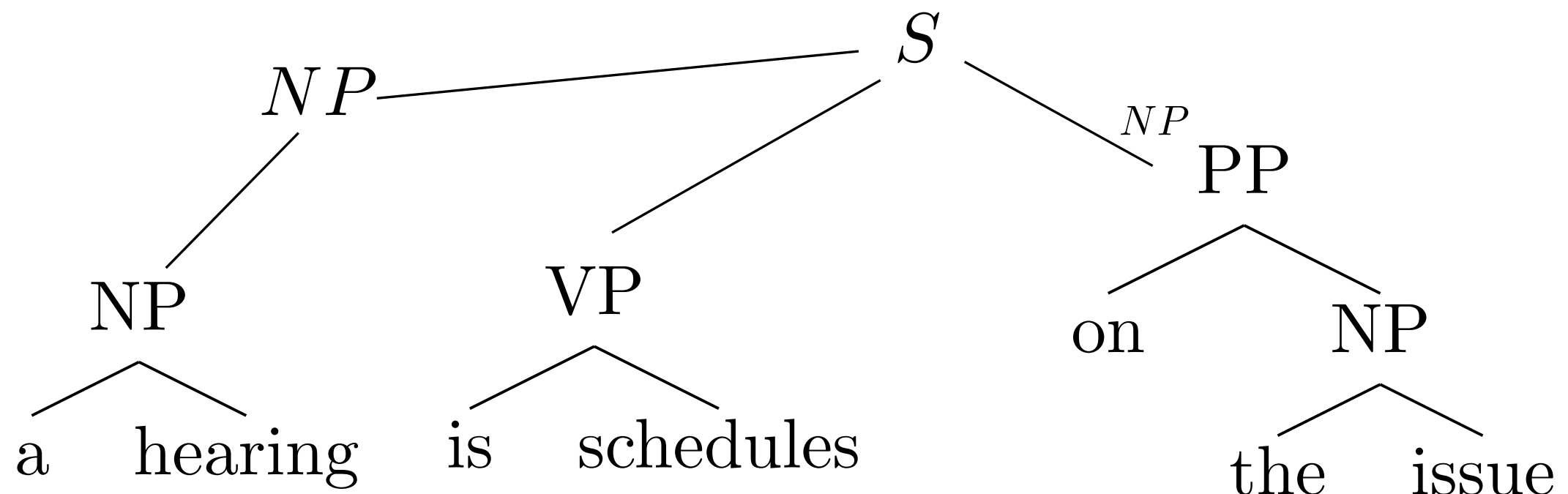
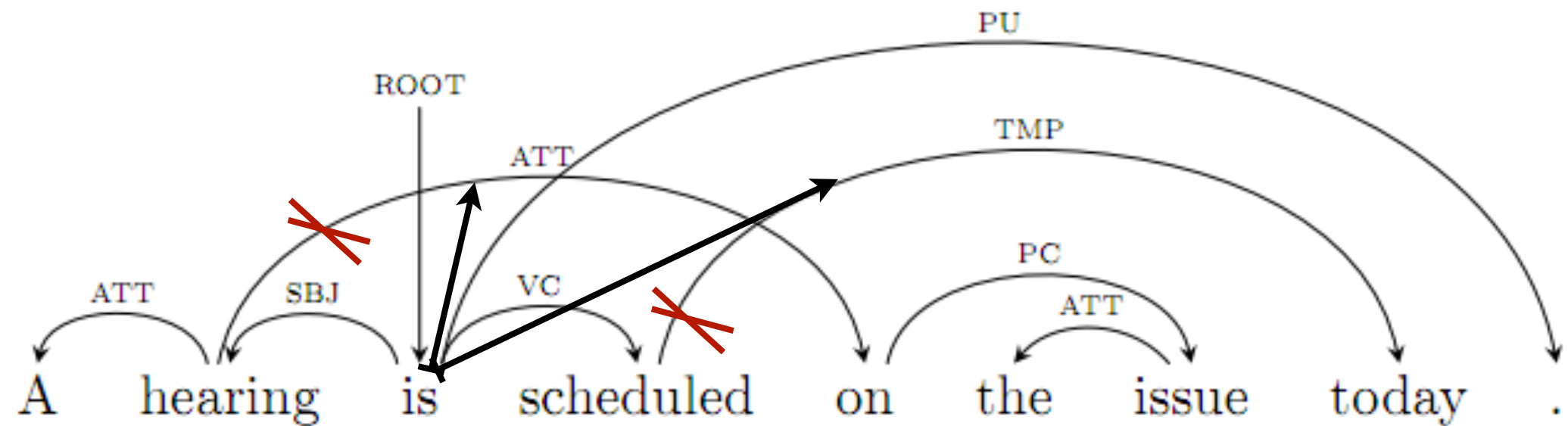
A Comment on Nonprojectivity



A Comment on Nonprojectivity



Pseudo projectivity



Attachment Scores

CoNLL Shared Tasks 2006 and 2007

Attachment Scores

CoNLL Shared Tasks 2006 and 2007

- Turns trees T , G into sets
- Measure the sets' size
- Use the intersection

Attachment Scores

CoNLL Shared Tasks 2006 and 2007

- Turns trees T , G into sets
- Measure the sets' size
- Use the intersection

$$Precision(T, G) = \frac{|T \cap G|}{|T|}$$

$$Recall(T, G) = \frac{|T \cap G|}{|G|}$$

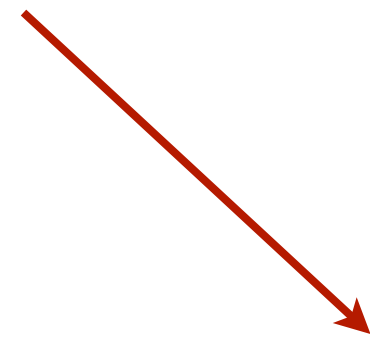
$$Fscore(T, G) = \frac{2 \times P \times R}{P + R}$$

Attachment Scores

CoNLL Shared Tasks 2006 and 2007

- Turns trees T , G into sets
- Measure the sets' size
- Use the intersection

Arcs of T



$$Precision(A_T, G) = \frac{|A_T \cap G|}{|A_T|}$$

$$Recall(A_T, G) = \frac{|A_T \cap G|}{|G|}$$

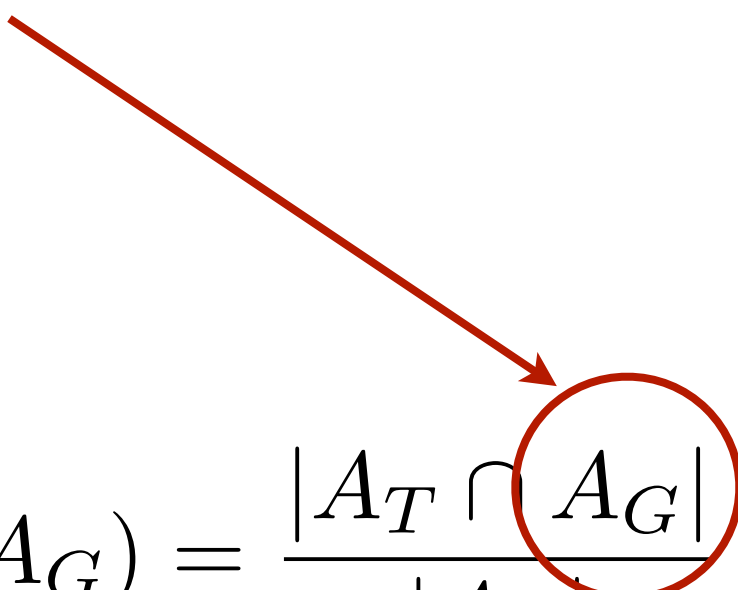
$$Fscore(A_T, G) = \frac{2 \times P \times R}{P + R}$$

Attachment Scores

CoNLL Shared Tasks 2006 and 2007

- Turns trees T , G into sets
- Measure the sets' size
- Use the intersection

Arcs of G


$$Precision(A_T, A_G) = \frac{|A_T \cap A_G|}{|A_T|}$$

$$Recall(A_T, A_G) = \frac{|A_T \cap A_G|}{|A_G|}$$

$$Fscore(A_T, A_G) = \frac{2 \times P \times R}{P + R}$$

Attachment Scores

CoNLL Shared Task 2006 and 2007

- Turns trees T , G into sets
- Measure the sets' size
- Use the intersection

$$Precision(A_T, A_G) = \frac{|A_T \cap A_G|}{|A_T|}$$

$$Recall(A_T, A_G) = \frac{|A_T \cap A_G|}{|A_G|}$$

$$Fscore(A_T, A_G) = \frac{|A_T \cap A_G|}{|S|}$$

Attachment Scores

CoNLL Shared Task 2006 and 2007

- Turns trees T , G into sets
- Measure the sets' size
- Use the intersection

$$Precision(A_T, A_G) = \frac{|A_T \cap A_G|}{|A_T|}$$

$$Recall(A_T, A_G) = \frac{|A_T \cap A_G|}{|A_G|}$$

$$Fscore(A_T, A_G) = \frac{|A_T \cap A_G|}{|S|}$$

Attachment Scores

CoNLL Shared Task 2006 and 2007

- Turns trees T , G into sets
- Measure the sets' size
- Use the intersection

$$Precision(A_T, A_G) = \frac{|A_T \cap A_G|}{|A_T|}$$

$$Recall(A_T, A_G) = \frac{|A_T \cap A_G|}{|A_G|}$$

UAS (unlabeled)

$$Fscore(A_T, A_G) = \frac{|A_T \cap A_G|}{|S|}$$

Attachment Scores

CoNLL Shared Task 2006 and 2007

- Turns trees T , G into sets
- Measure the sets' size
- Use the intersection

$$Precision(A_T, A_G) = \frac{|A_T \cap A_G|}{|A_T|}$$

LAS (labeled)

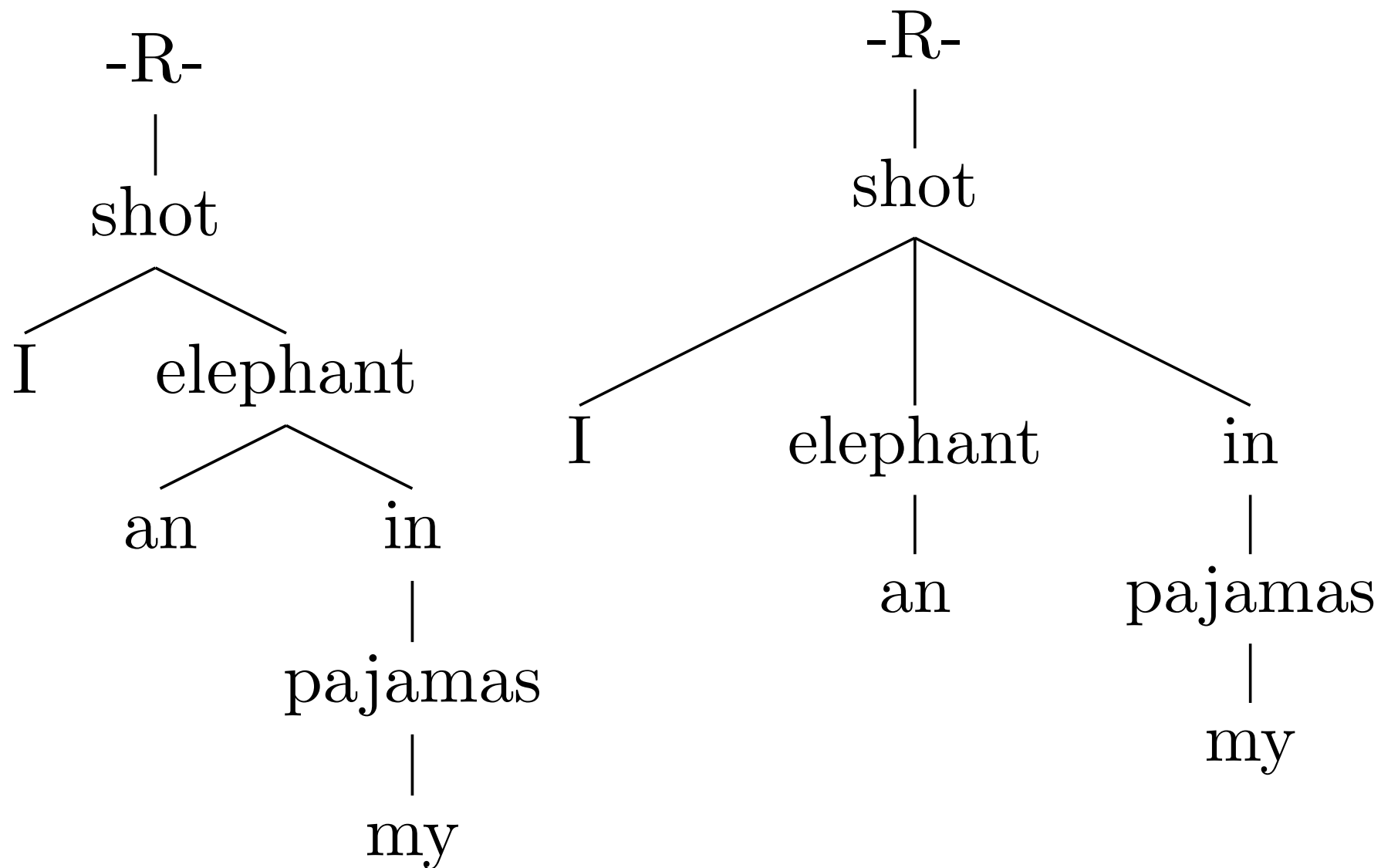
$$Recall(A_T, A_G) = \frac{|A_T \cap A_G|}{|A_G|}$$

UAS (unlabeled)

$$Fscore(A_T, A_G) = \frac{|A_T \cap A_G|}{|S|}$$

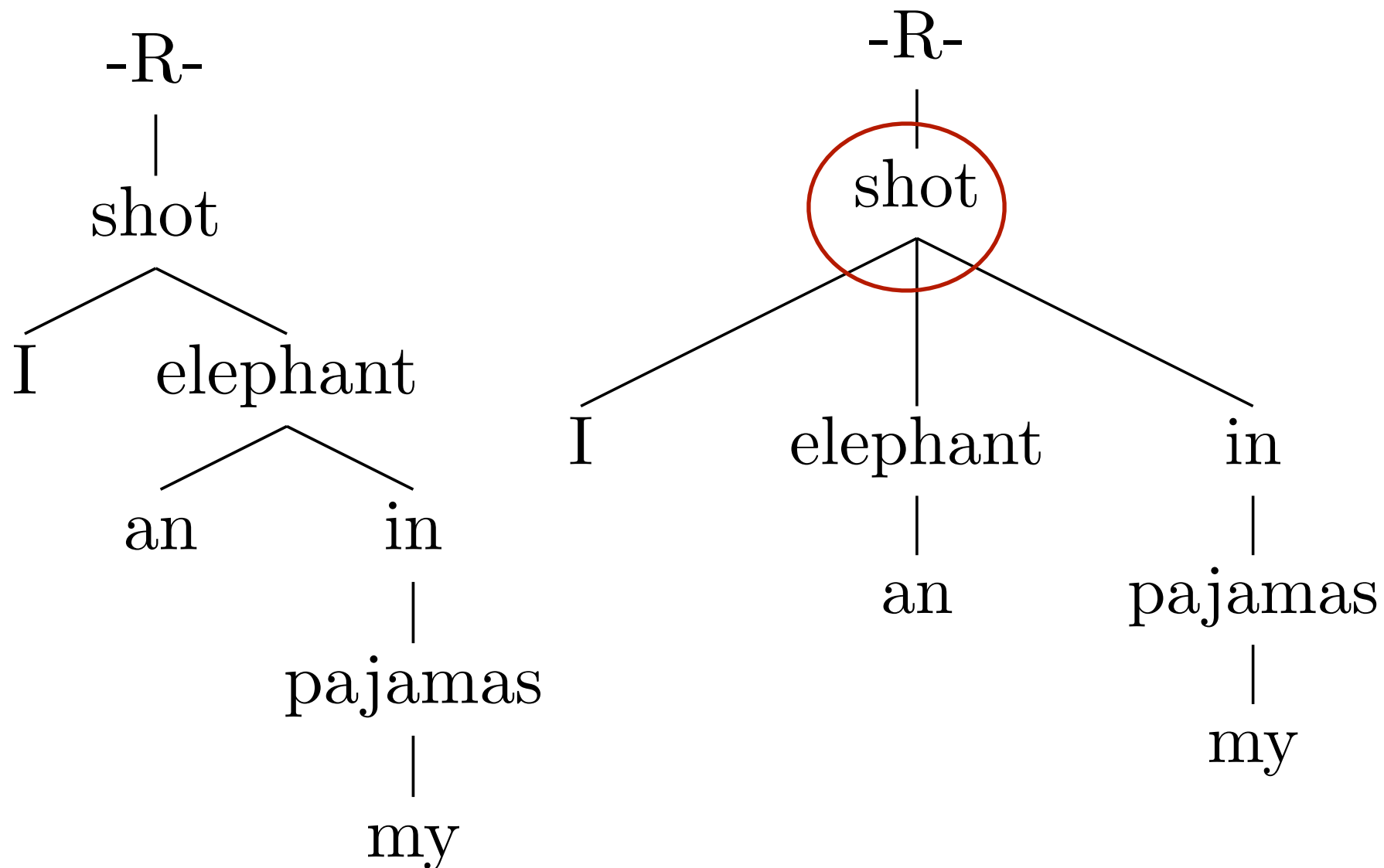
Attachment Scores

CoNLL Shared Task 2006 and 2007



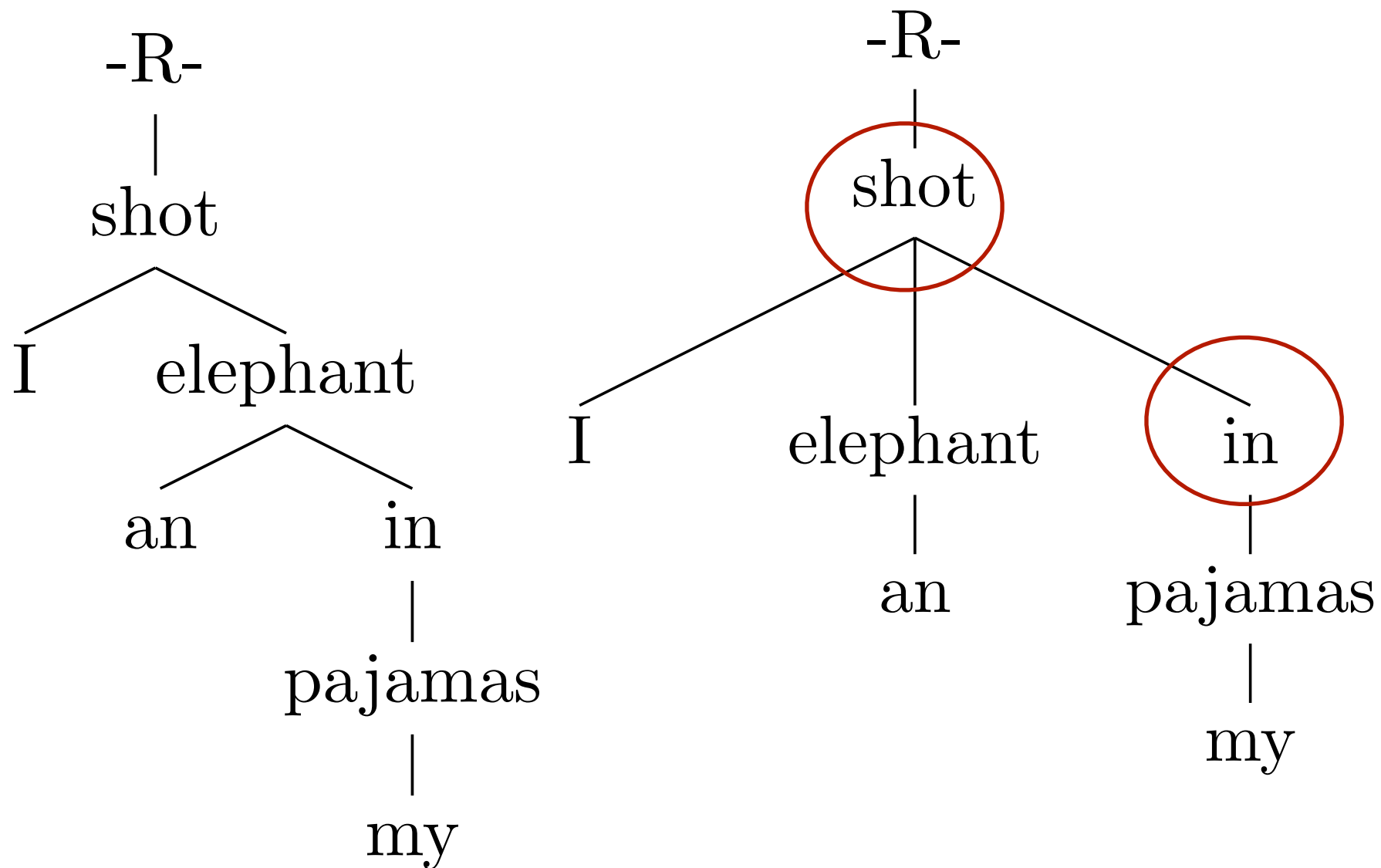
Attachment Scores

CoNLL Shared Task 2006 and 2007



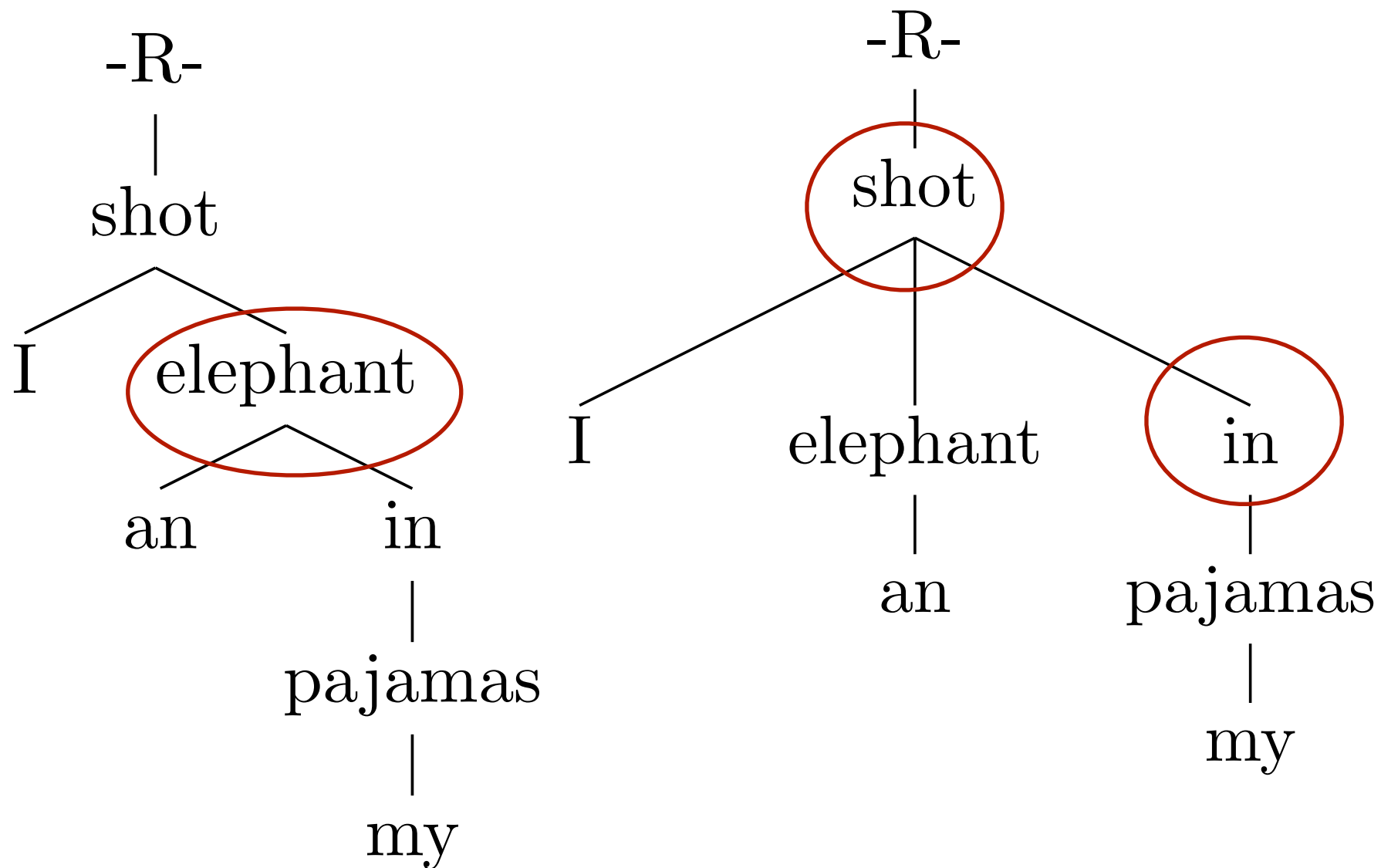
Attachment Scores

CoNLL Shared Task 2006 and 2007



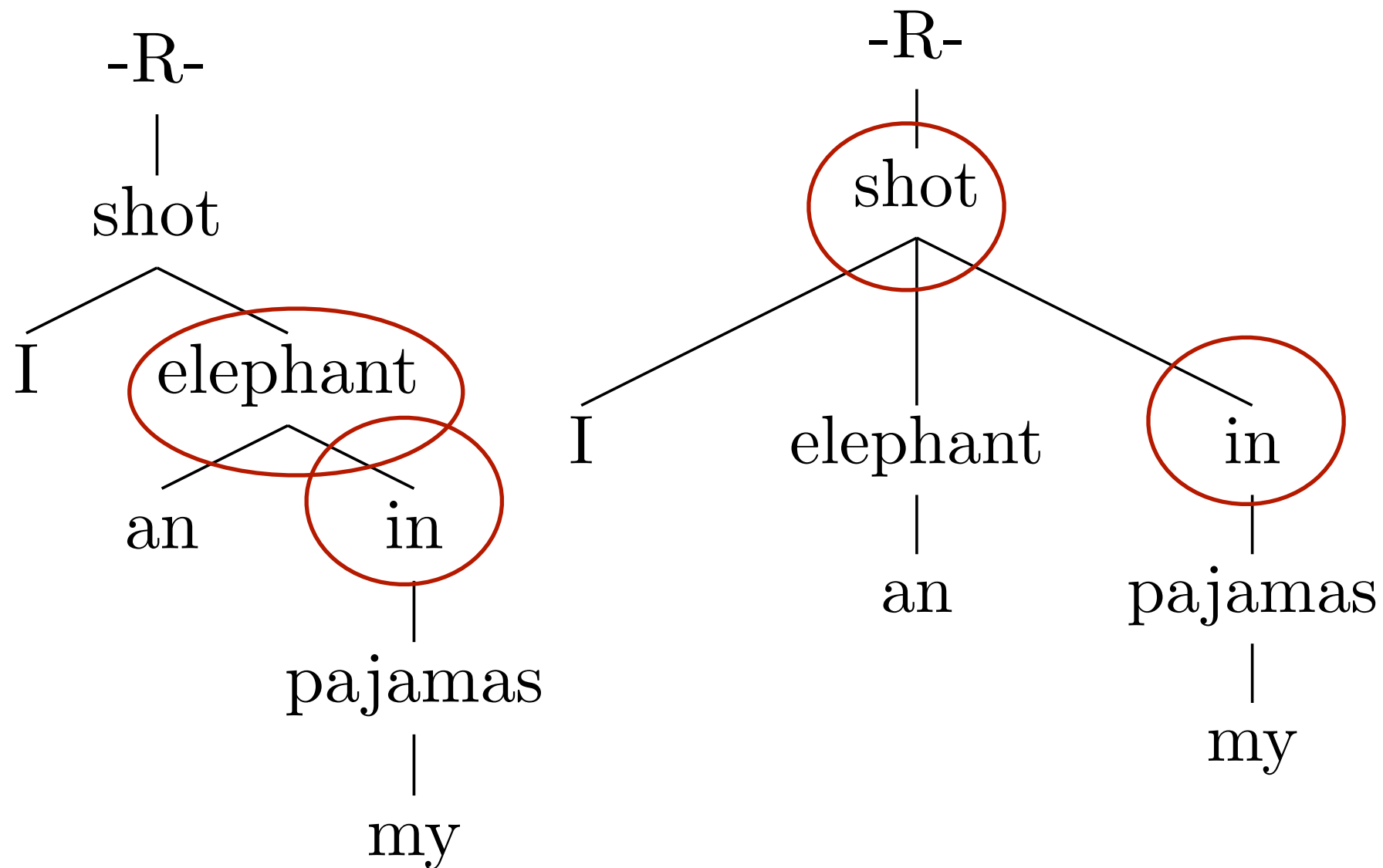
Attachment Scores

CoNLL Shared Task 2006 and 2007



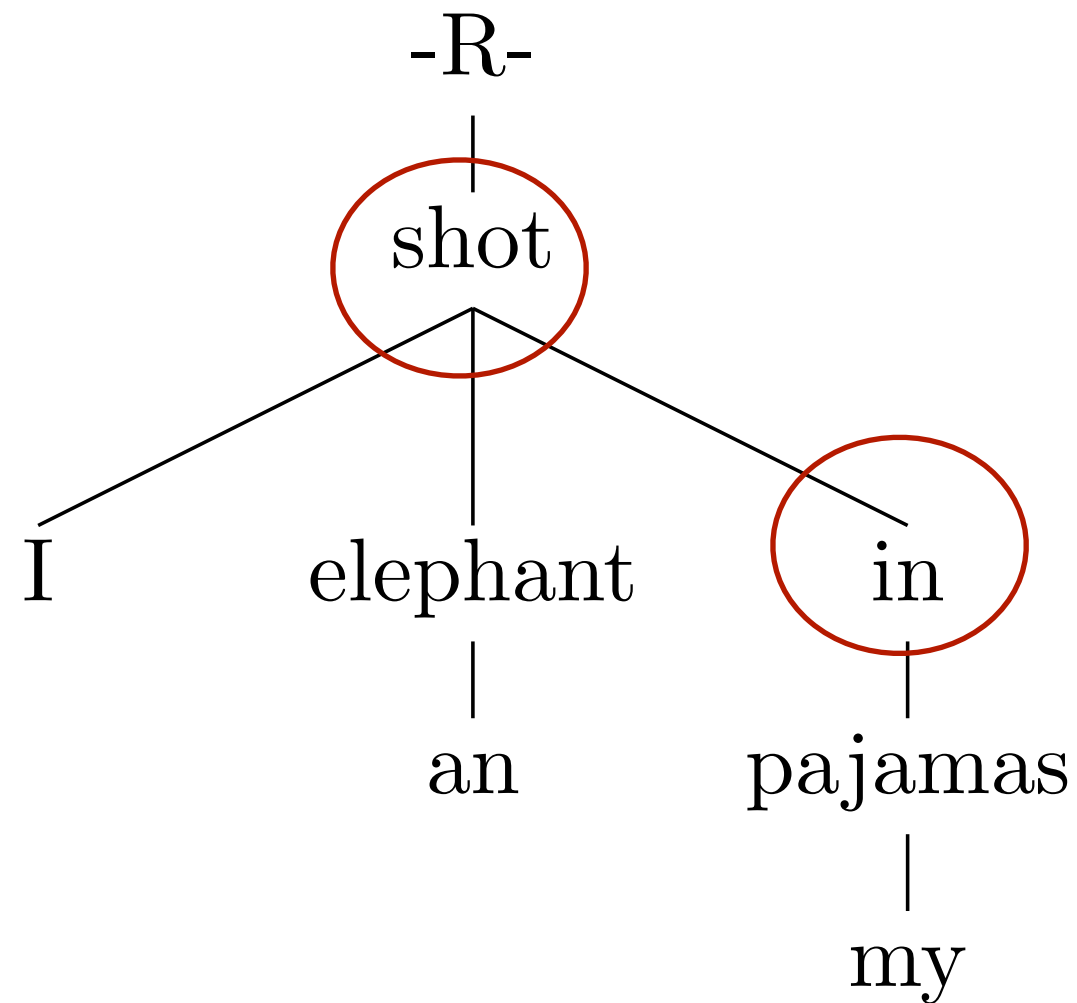
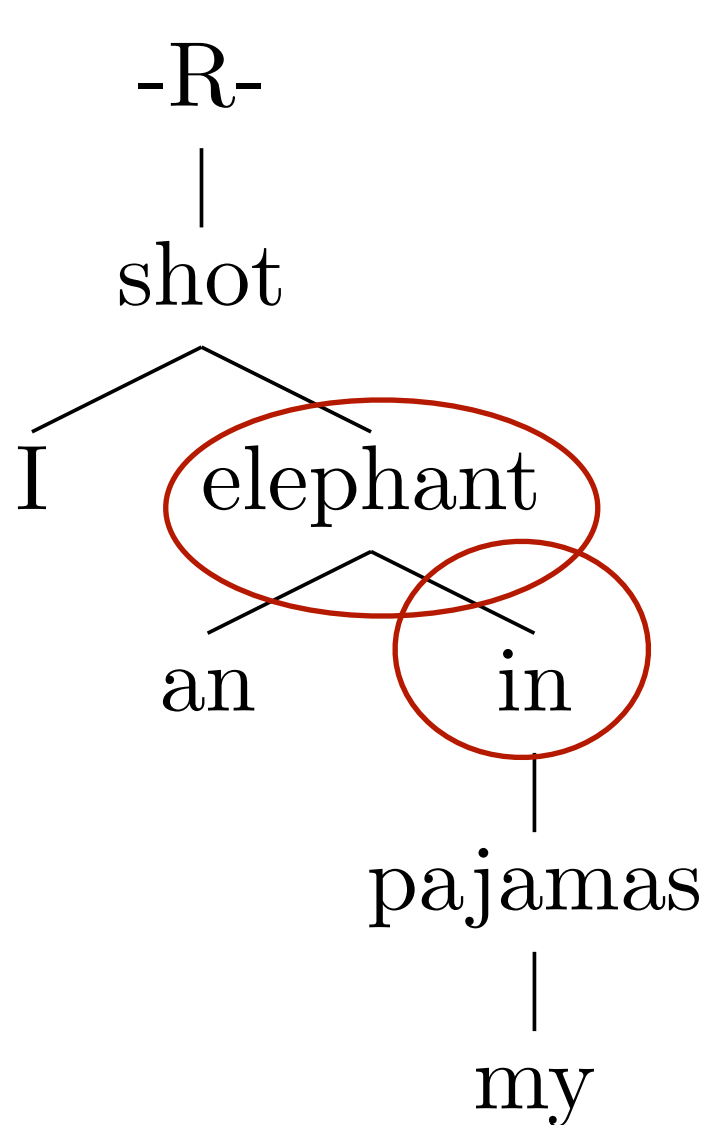
Attachment Scores

CoNLL Shared Task 2006 and 2007



Attachment Scores

CoNLL Shared Task 2006 and 2007



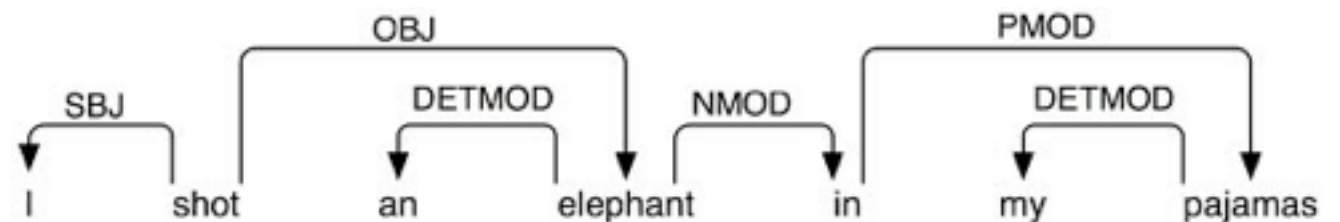
$$UAS(A_G, A_T) = \frac{|A_G| \cap |A_T|}{7} = \frac{6}{7}$$

Architectural Decisions

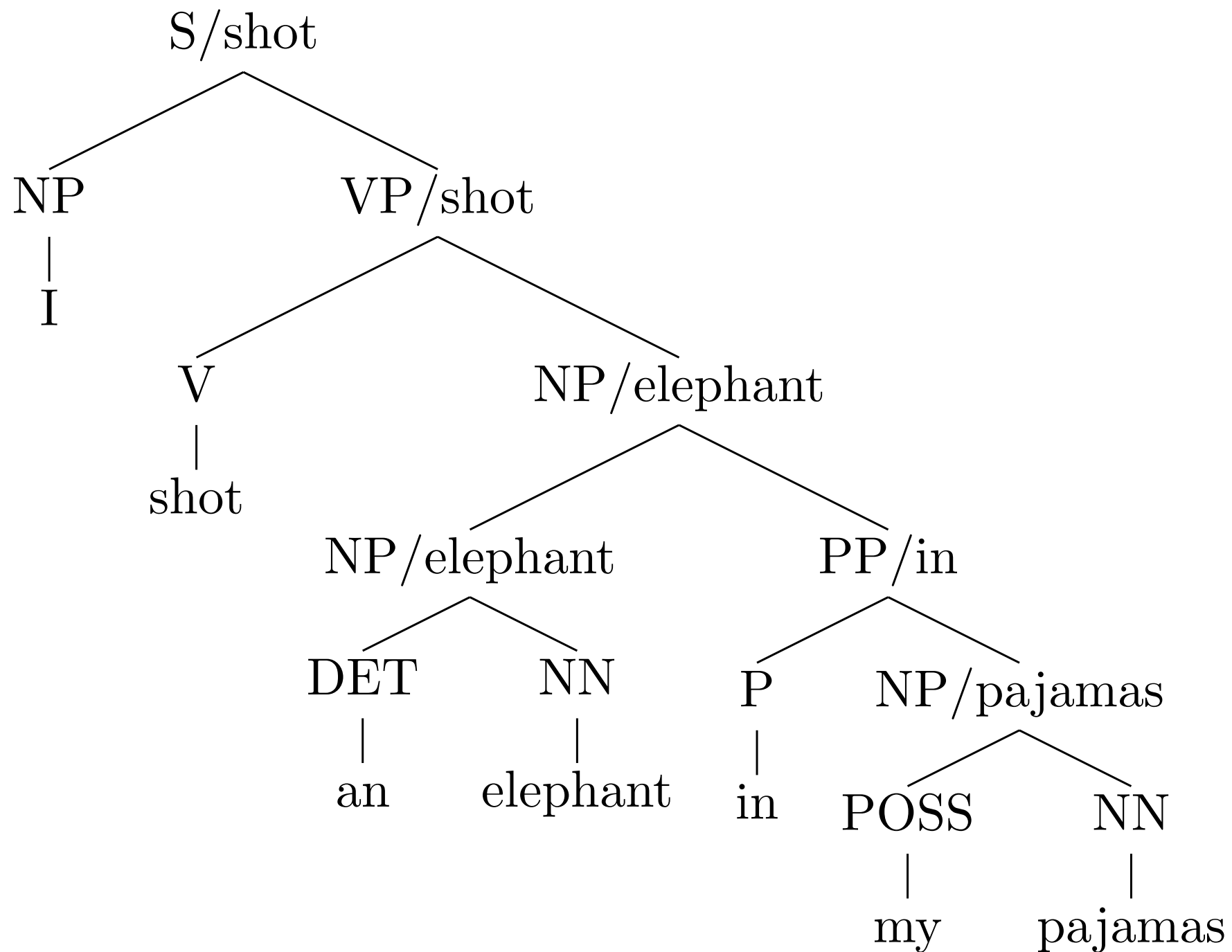
- Representation: Dependency Trees
- Model: ?
- Inference: ?
- Learning: ?
- Evaluation: Labeled/Unlabeled AS

Models for DP

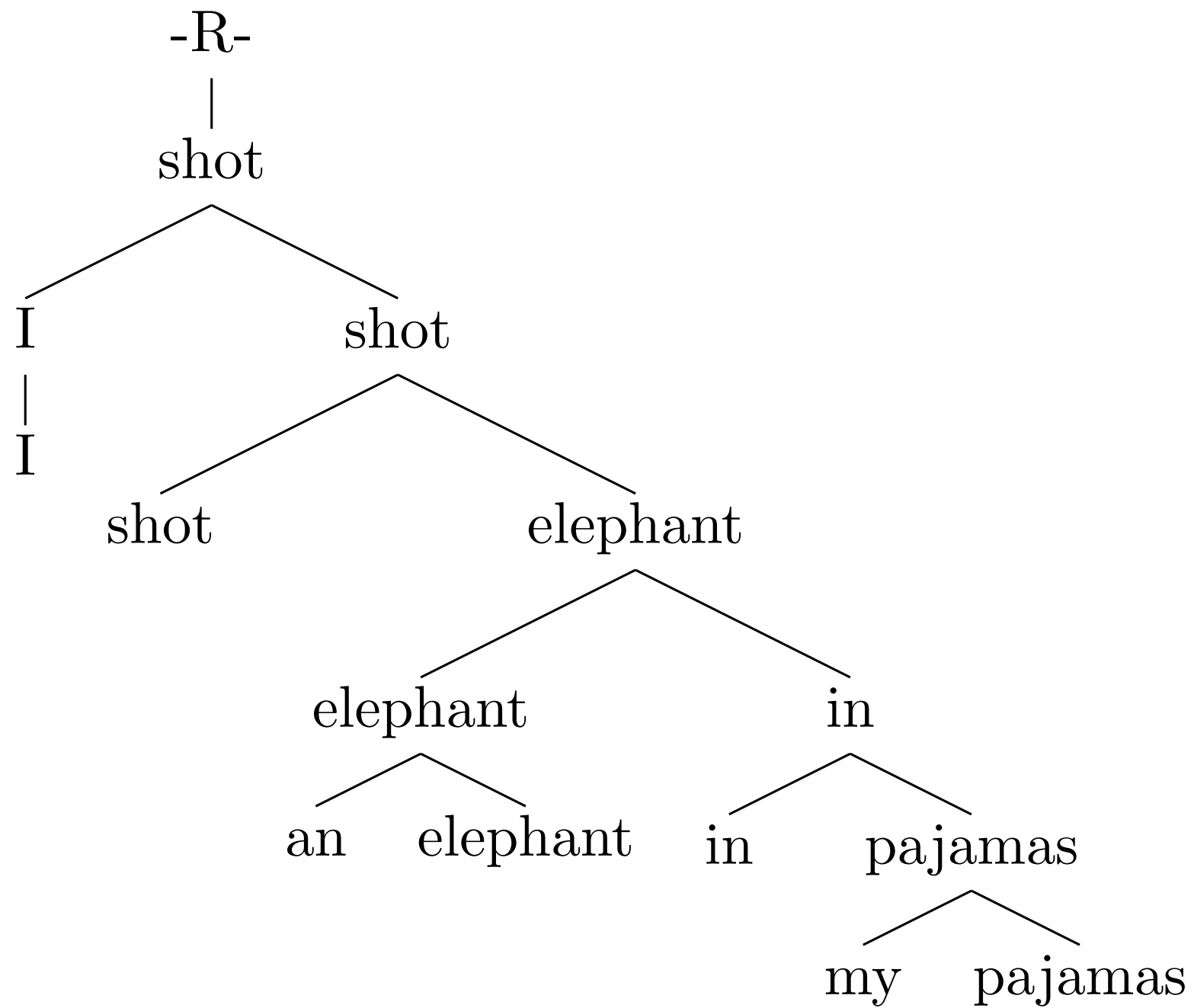
- Modeling Techniques
 - Grammar-Based
 - Graph-Based
 - Transition-Based



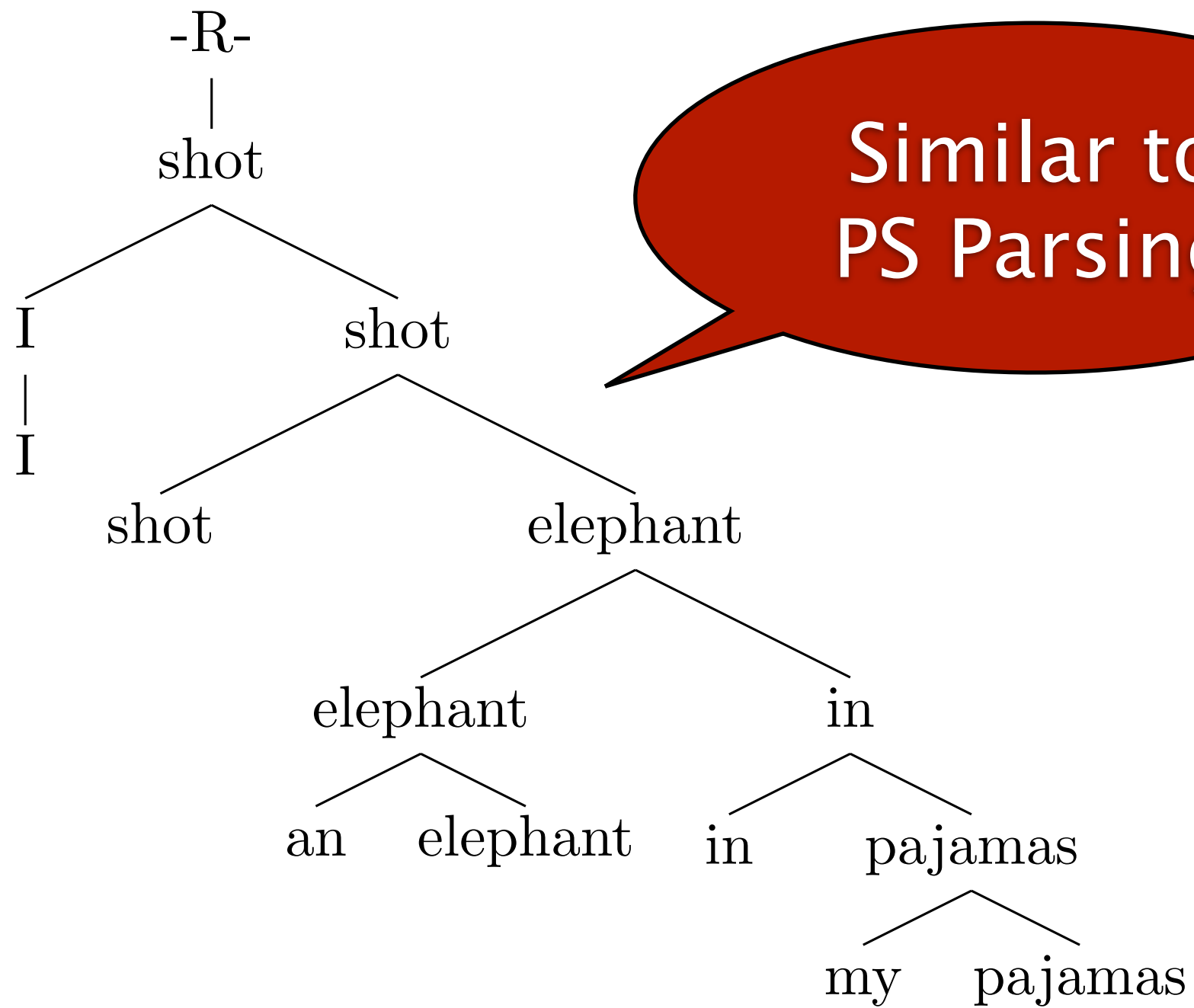
(1) Grammar-Based



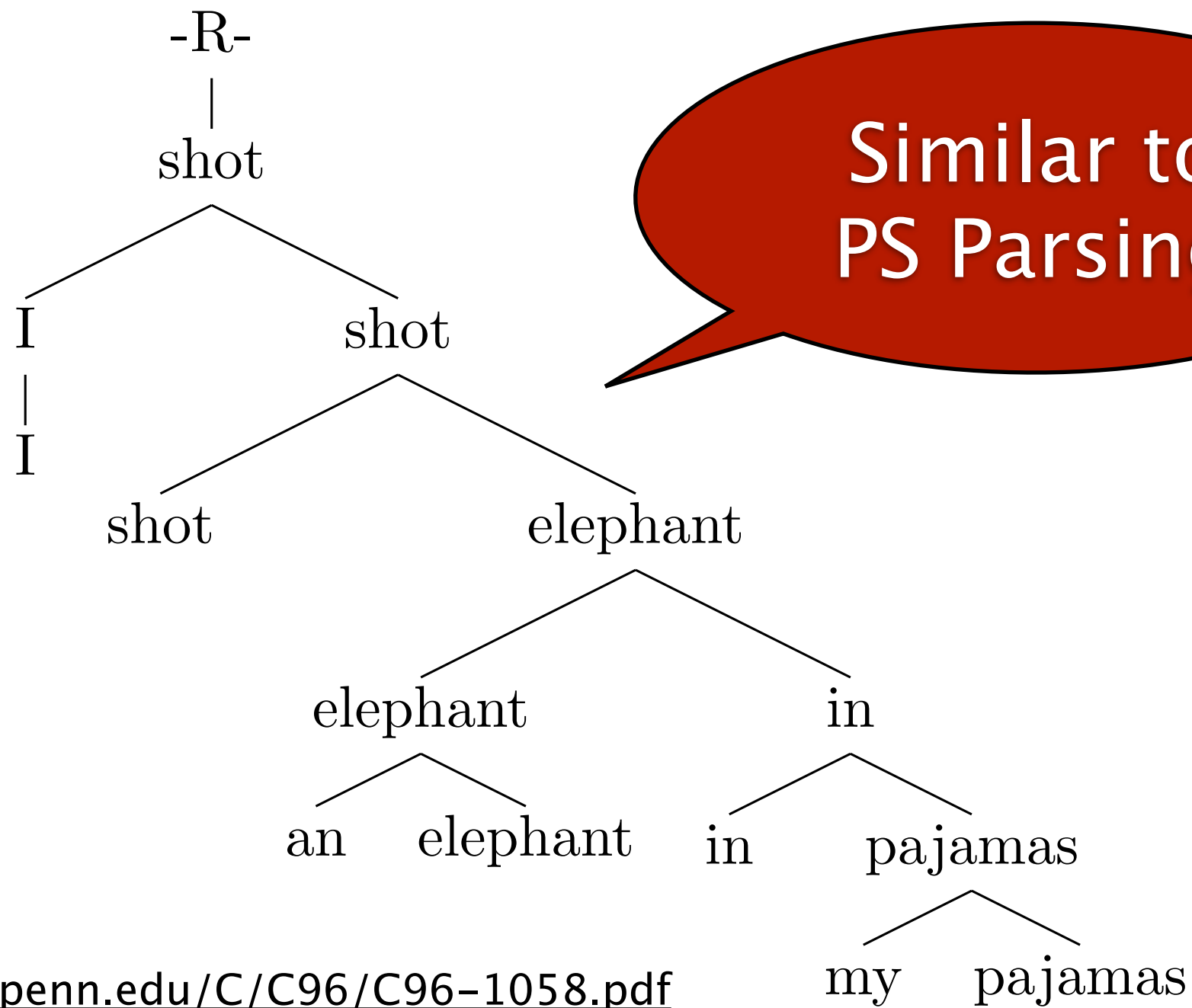
(1) Grammar-Based



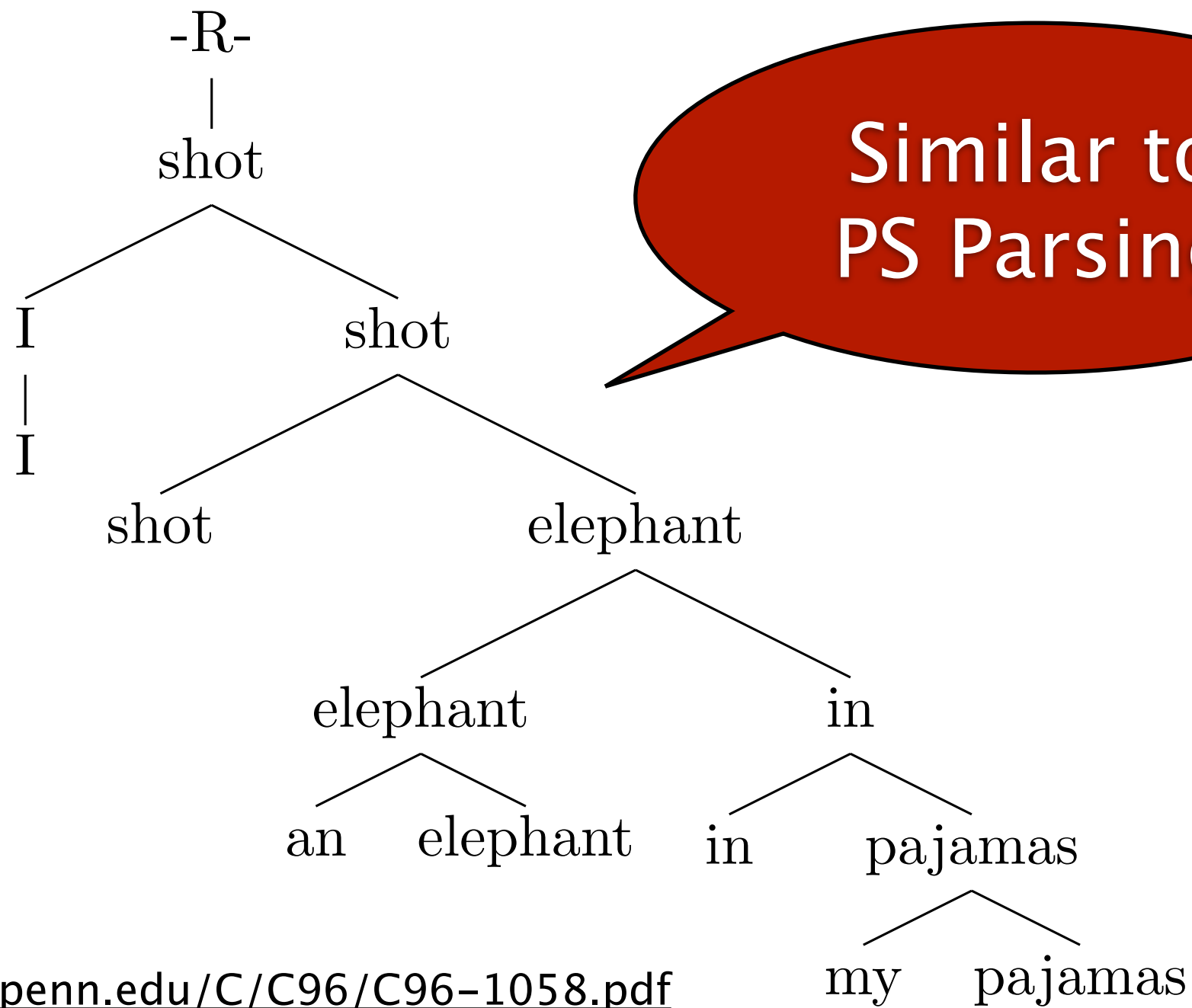
(1) Grammar-Based



(1) Grammar-Based



✓ (1) Grammar-Based



(2) Graph-Based

(2) Graph-Based

- The Idea:
 - Treat the input sentence as a bag of words (each word is a node)
 - Use available graph-algorithms to find a spanning tree on the nodes

(2) Graph-Based

-R-

I

pajamas

shot

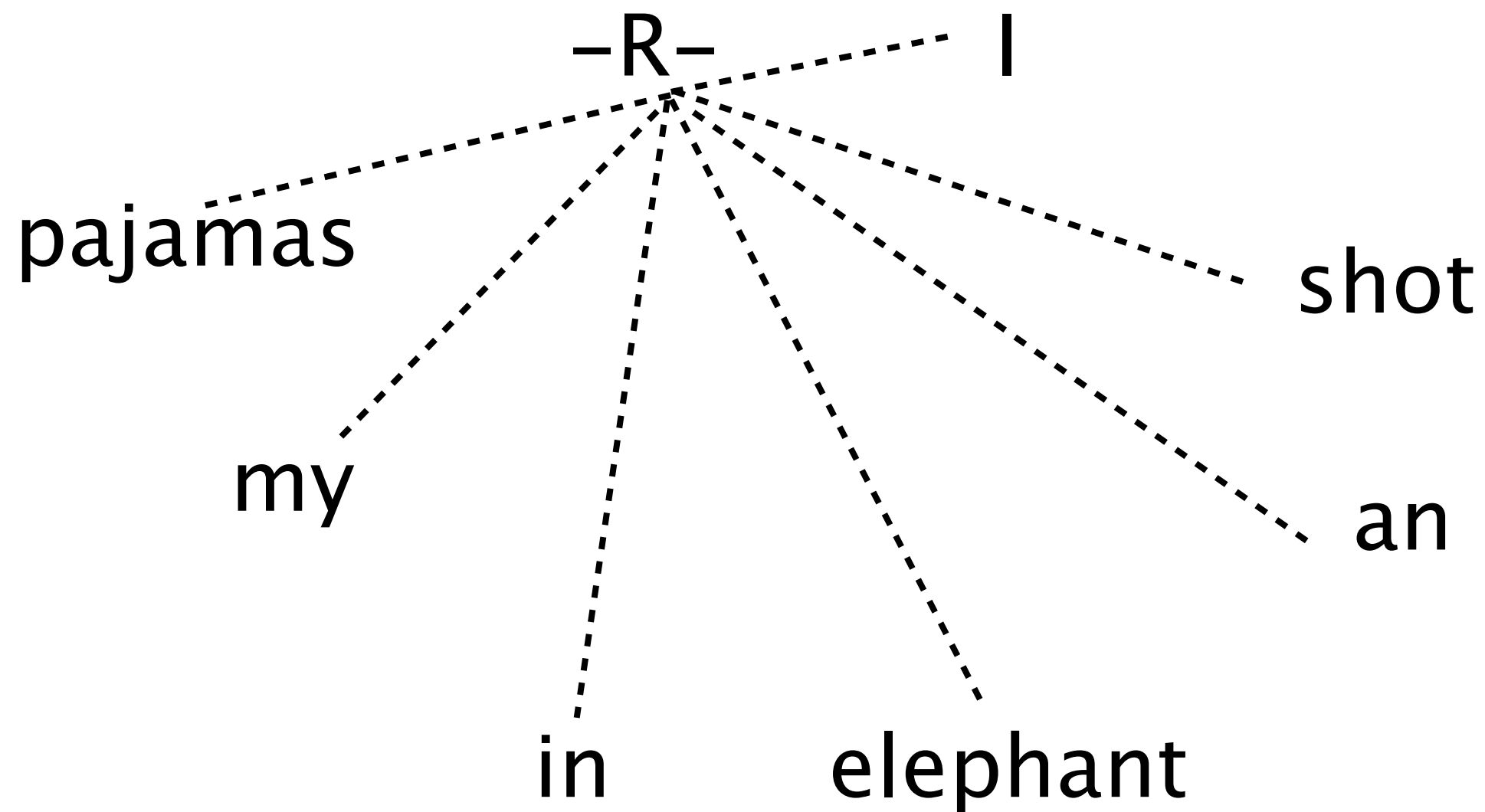
my

an

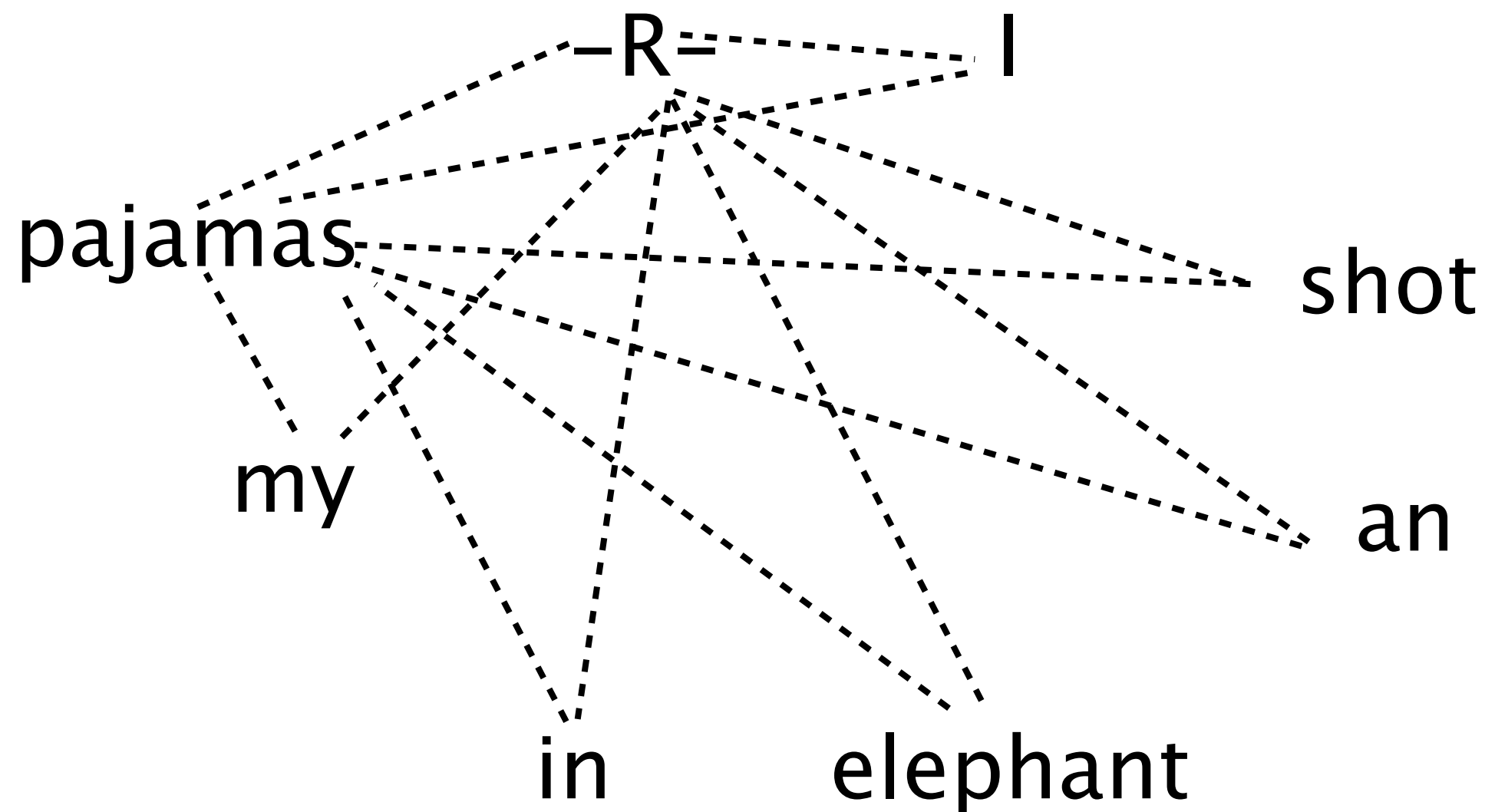
in

elephant

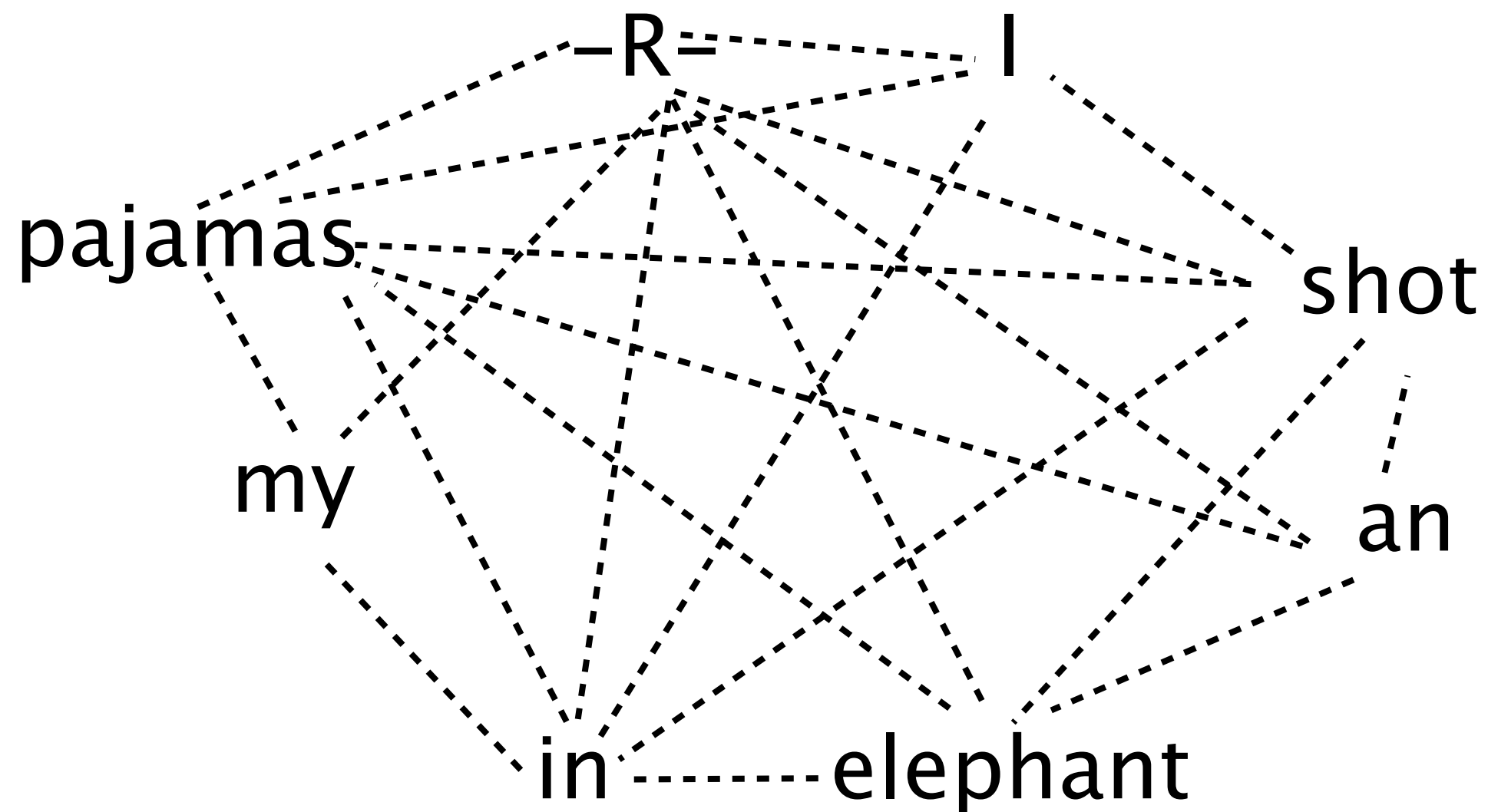
(2) Graph-Based



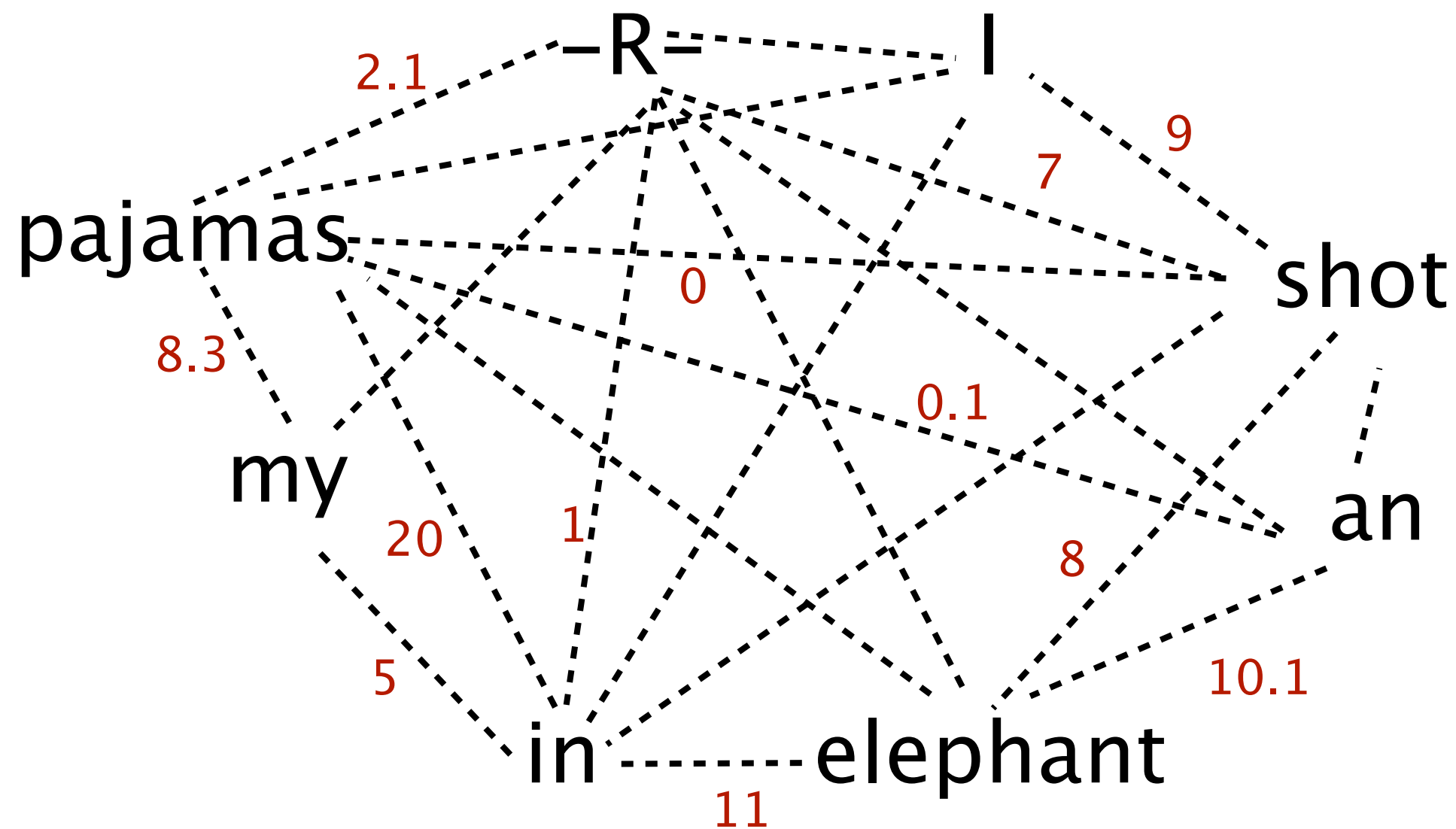
(2) Graph-Based



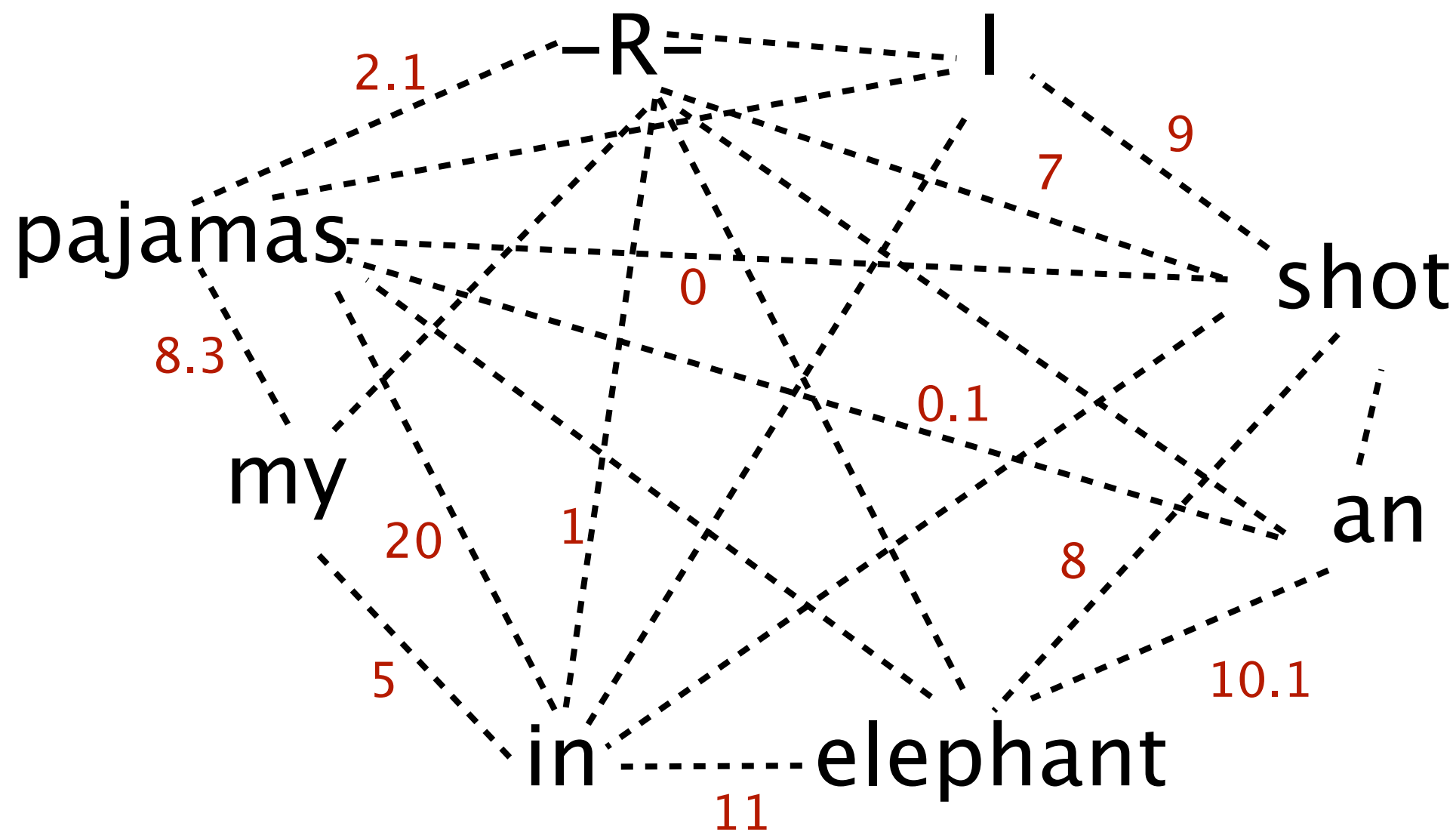
(2) Graph-Based



(2) Graph-Based

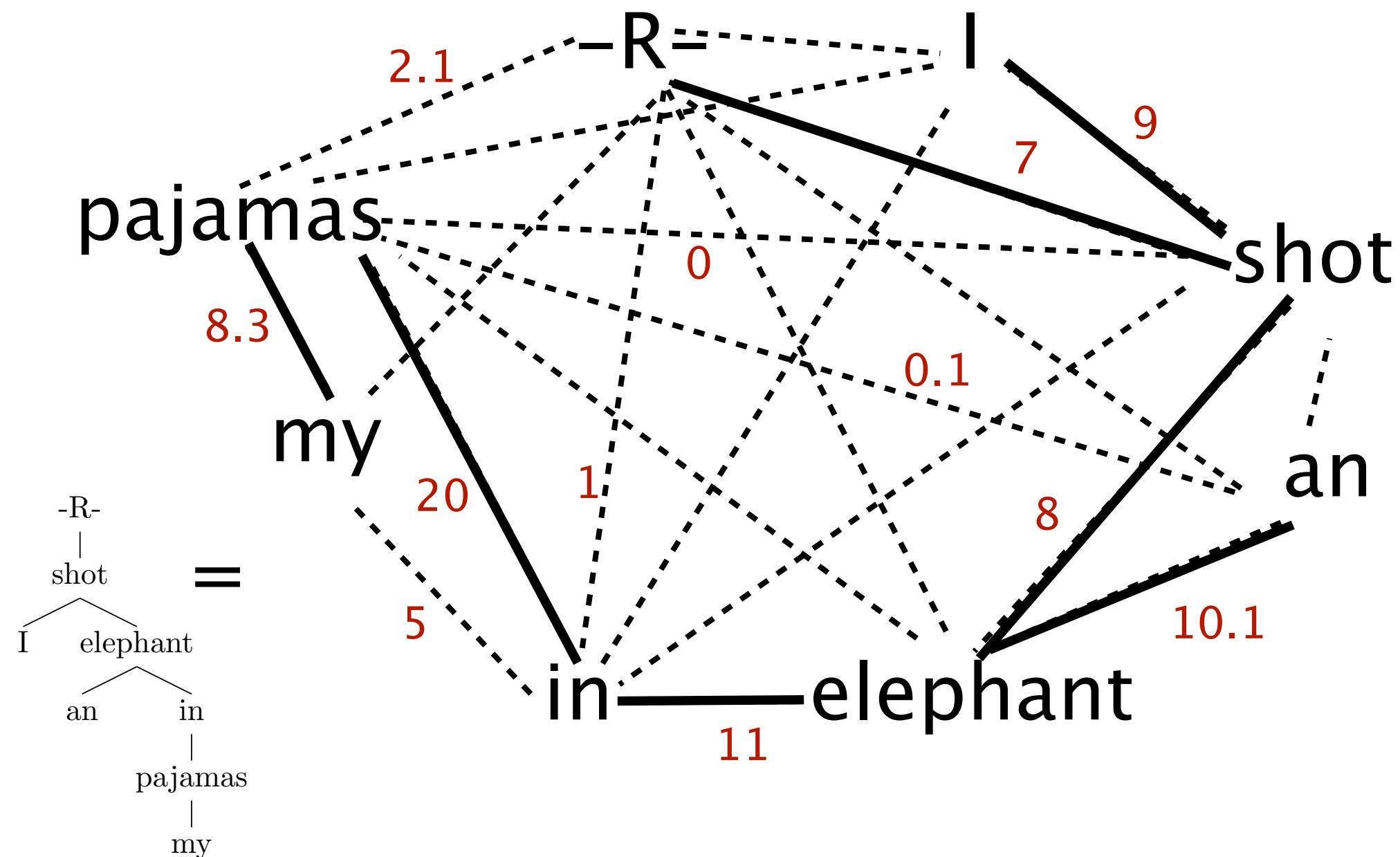


(2) Graph-Based



Goal: Find the Maximum Spanning Tree (MST)

(2) Graph-Based



(2) Graph-Based

- Graph-Based Dependency Parsing
 - Inference:
 - Find the max-scoring tree
 - Learning:
 - Find the best scoring function

(2) Graph-Based

- Graph-Based Dependency Parsing
 - Inference:
 - Find the max-scoring tree
 - Learning:
 - Find the best scoring function



MIRA

(2) Graph-Based

- Graph-Based Dependency Parsing

- Inference:

- Find the max-scoring tree



MST

- Learning:

- Find the best scoring function



MIRA

(2) Graph-Based

- Graph-Based Dependency Parsing

- Inference:

- Find the max-scoring tree



MST

- Learning:

- Find the best scoring function



MIRA

<http://dl.acm.org/citation.cfm?id=1220641>

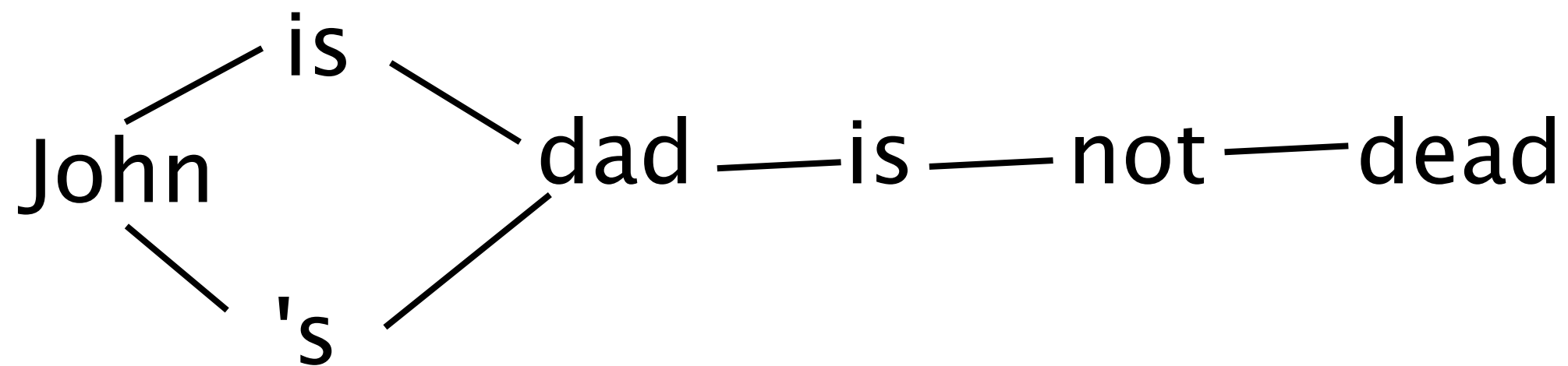
Graph-Based Methods for MRLs

Input Sentence:

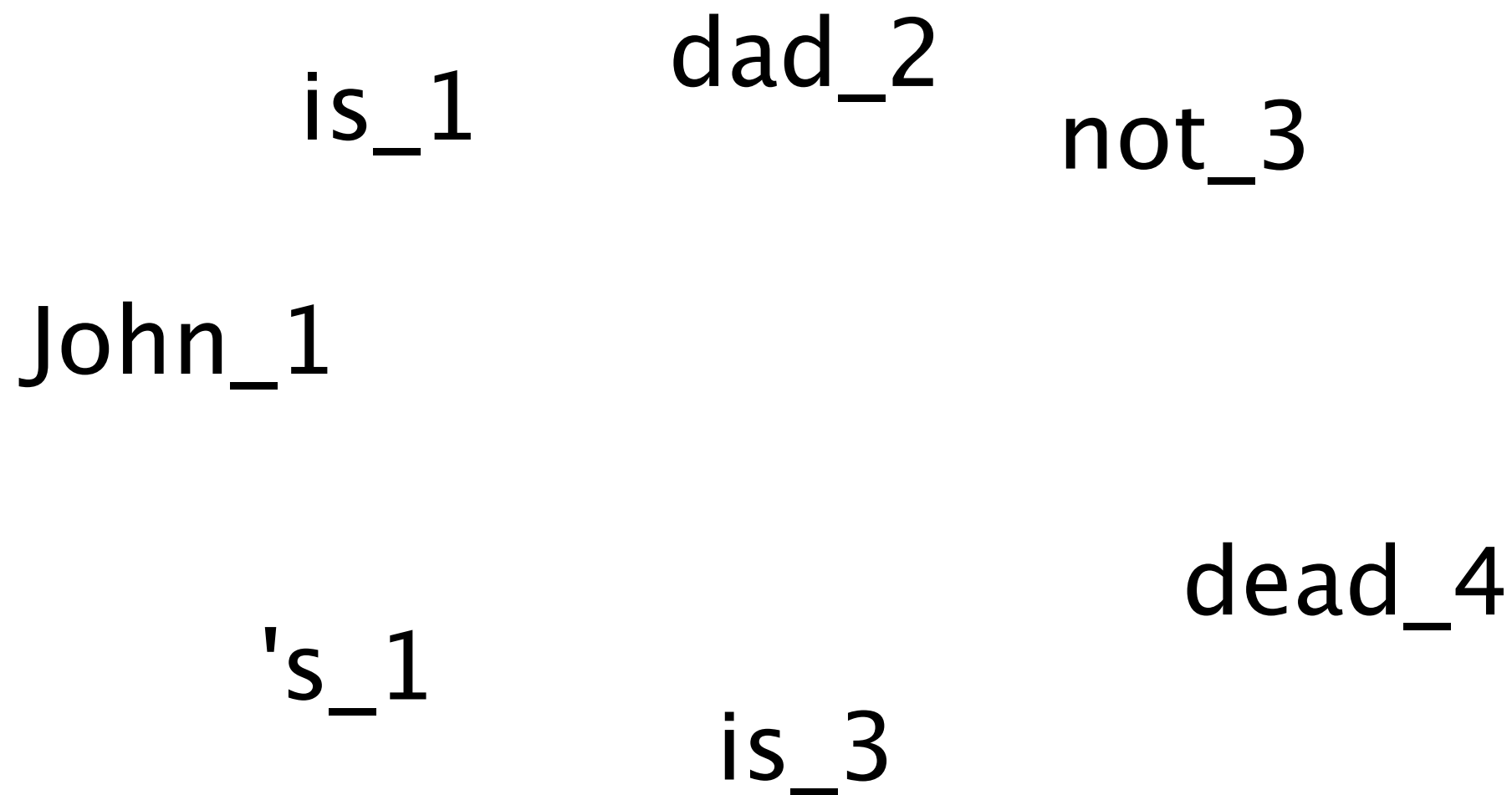
John's dad isn't dead

Graph-Based Methods for MRLs

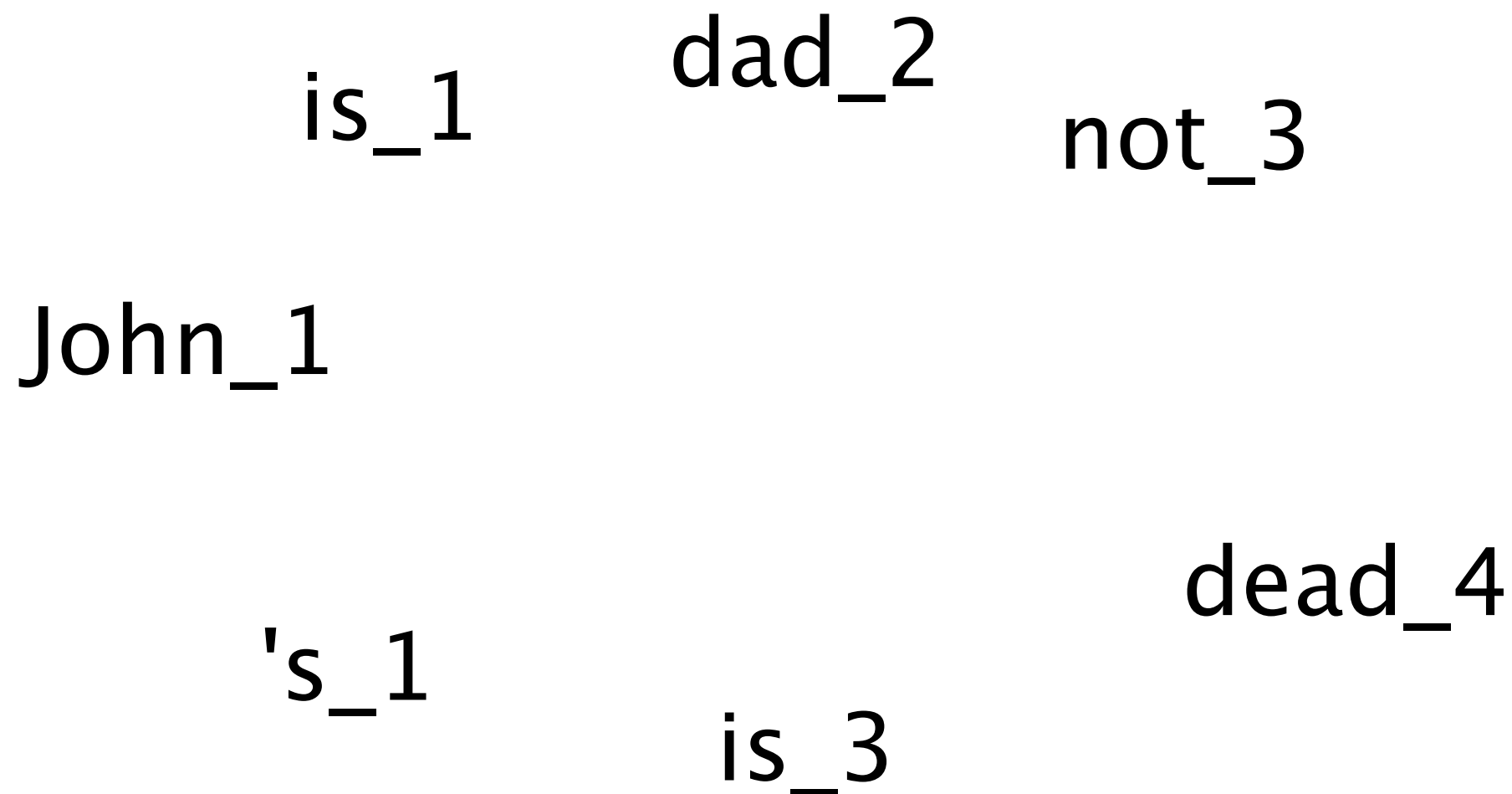
Input Lattice:



Graph-Based Methods for MRLs

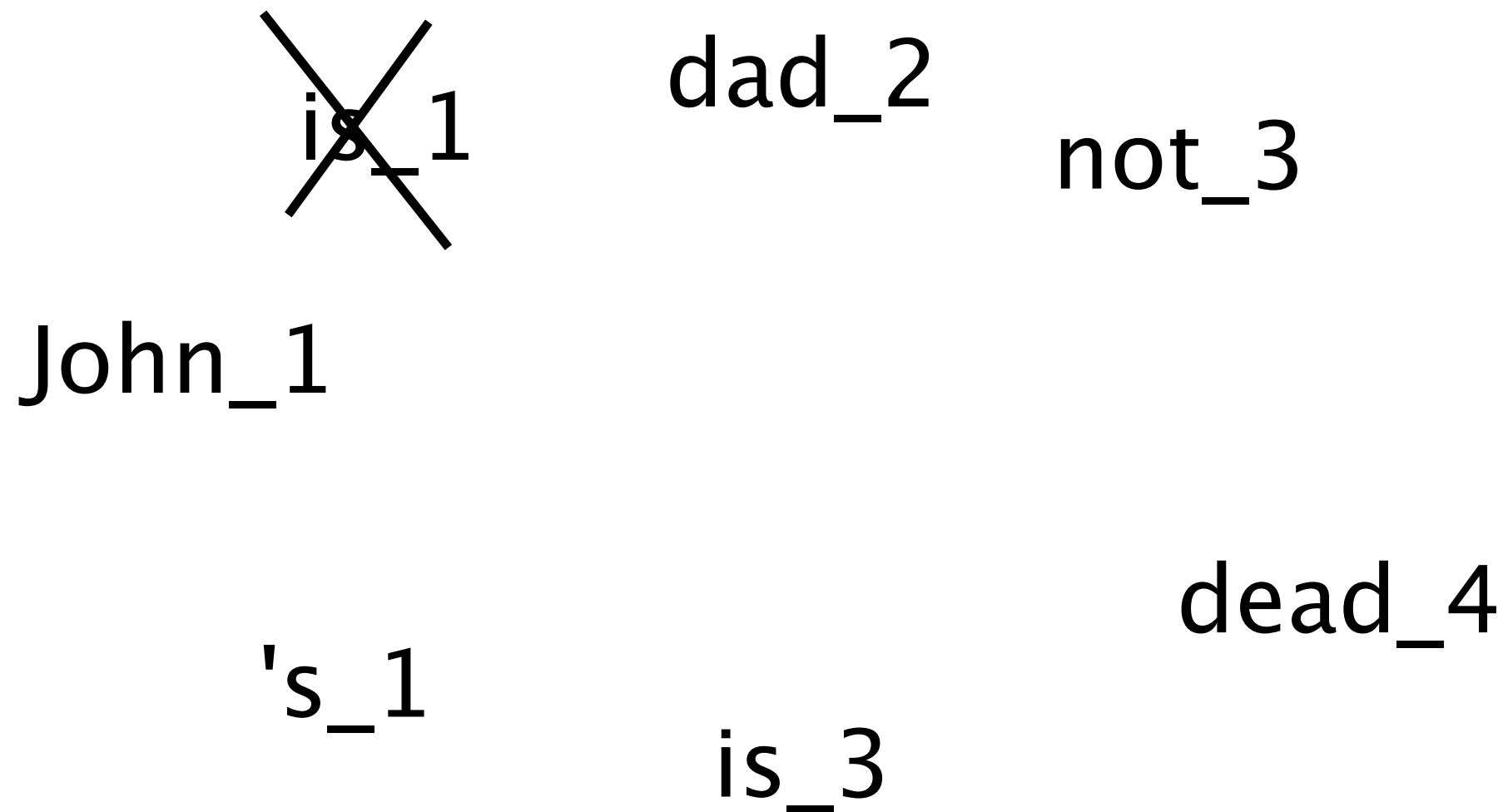


Graph-Based Methods for MRLs



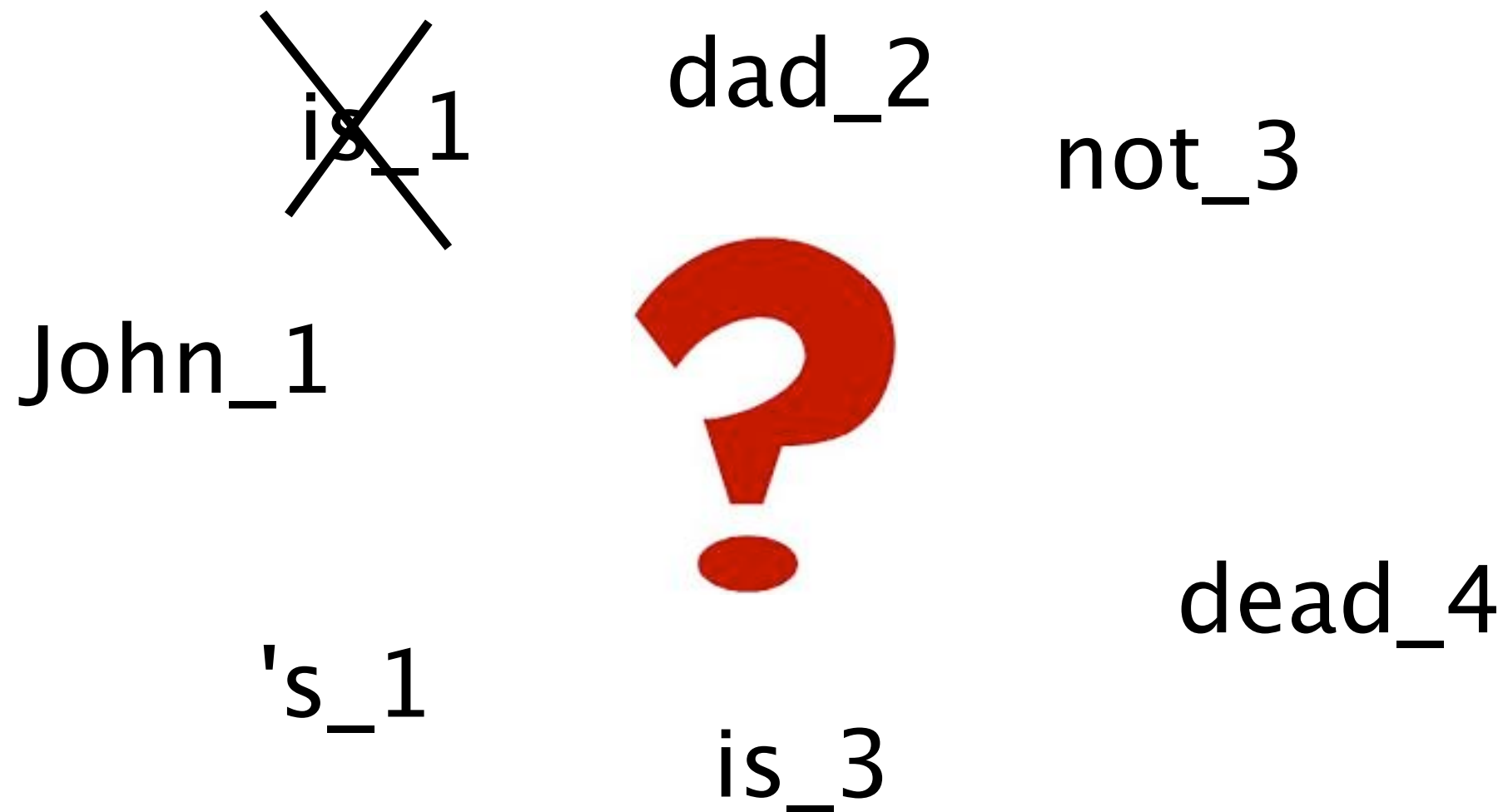
Goal: Find a Maximum Spanning tree
On the Best Subset of nodes

Graph-Based Methods for MRLs



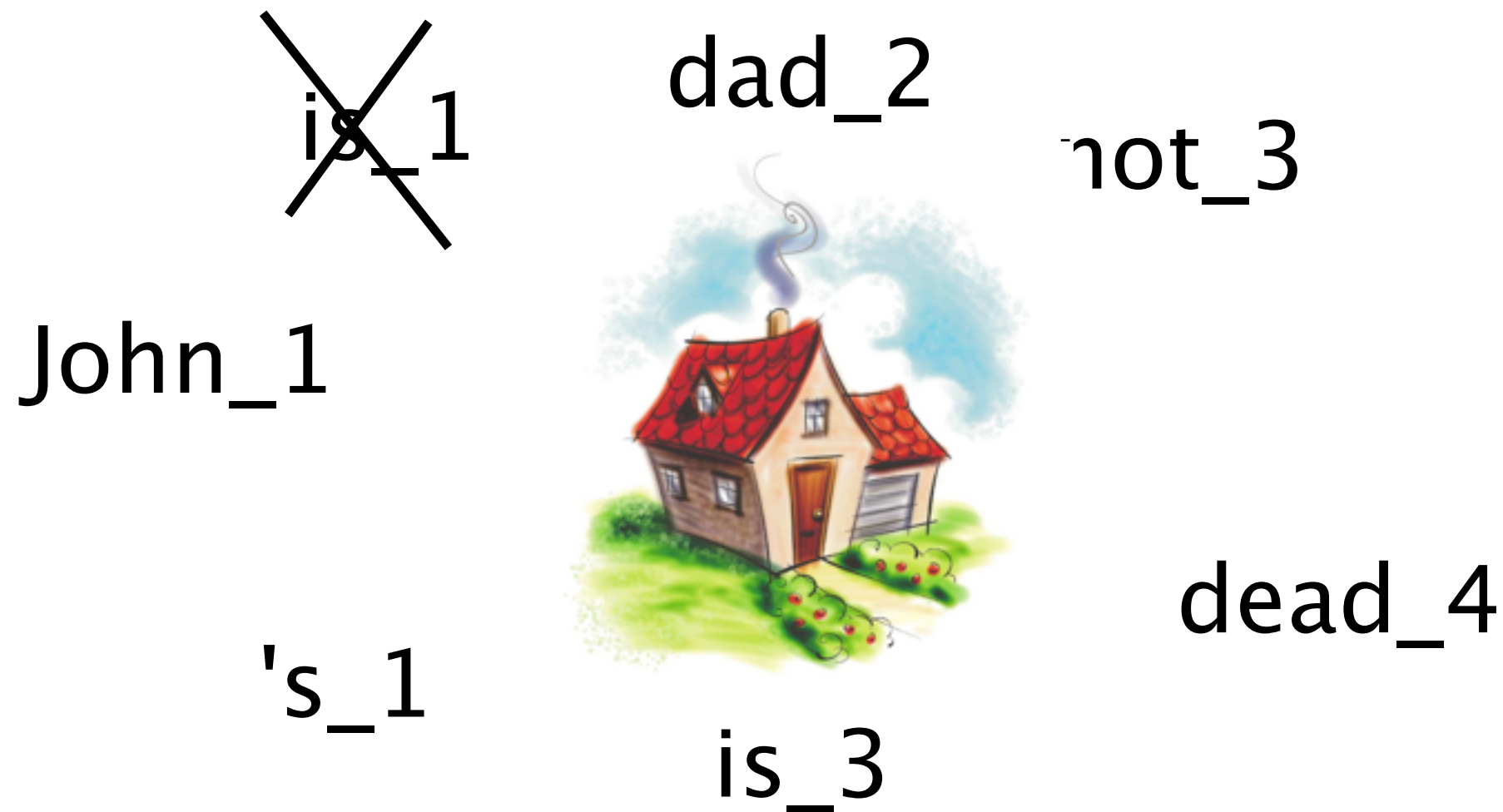
Goal: Find a Maximum Spanning tree
On the Best Subset of nodes

Graph-Based Methods for MRLs



Goal: Find a Maximum Spanning tree
On the Best Subset of nodes

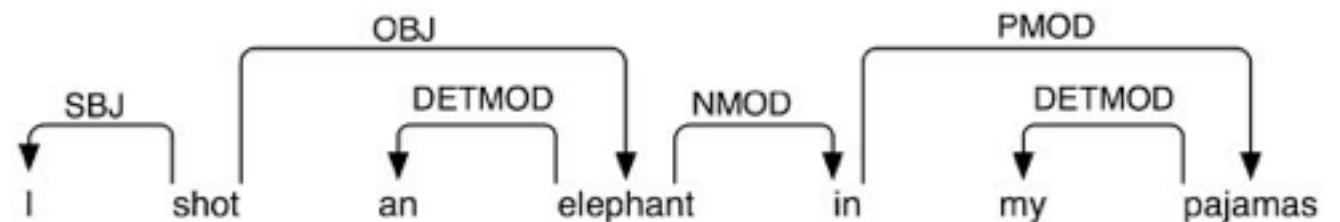
Graph-Based Methods for MRLs



Goal: Find a Maximum Spanning tree
On the Best Subset of nodes

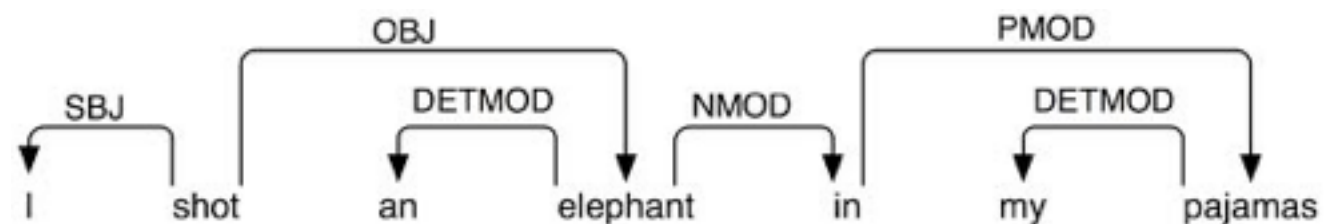
Models for DP

- Modeling Techniques
 - Grammar-Based reuse ideas from PS for MRLs
 - Graph-Based not yet worked out for MRLs
 - Transition-Based



Models for DP

- Modeling Techniques
 - Grammar-Based ✓ reuse ideas from PS for MRLs
 - Graph-Based not yet worked out for MRLs
 - Transition-Based



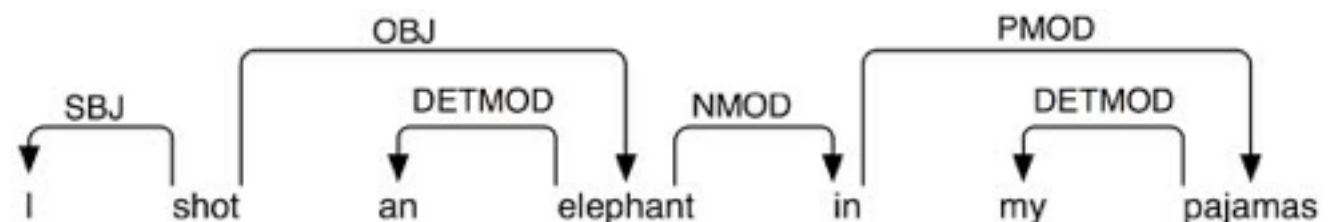
Models for DP

- Modeling Techniques

- Grammar-Based ✓ reuse ideas from PS for MRLs

- ~~● Graph Based~~ not yet worked out for MRLs

- Transition-Based



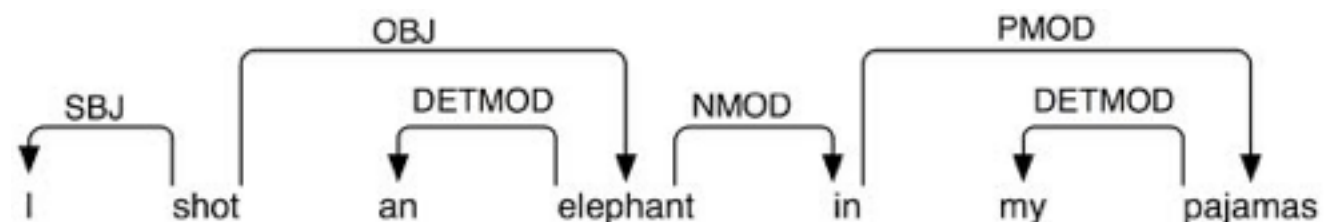
Models for DP

- Modeling Techniques

- Grammar-Based ✓ reuse ideas from PS for MRLs

- ~~● Graph Based~~ not yet worked out for MRLs

➔ Transition-Based



Models for DP

- Modeling Techniques
 - Grammar-Based
 - Graph-Based
 - Transition-Based



Questions
So Far?

Models for DP

- Modeling Techniques
 - Grammar-Based
 - Graph-Based

➔ Transition-Based



Questions
So Far?

Architectural Decisions

- Representation: Dependency Trees
- Model: Transition-Based
- Inference: ?
- Learning: ?
- Evaluation: Labeled/Unlabeled AS

Models for DS: Transition Systems

- A Transition system contains the following components
 - A Buffer
 - A Stack
 - A Set of Arcs

Models for DS: Transition Systems

- A Transition system contains the following components
 - A Buffer β
 - A Stack
 - A Set of Arcs

Models for DS: Transition Systems

- A Transition system contains the following components
 - A Buffer β
 - A Stack σ
 - A Set of Arcs

Models for DS: Transition Systems

- A Transition system contains the following components
 - A Buffer β
 - A Stack σ
 - A Set of Arcs A

Models for DS: Transition Systems

- A Transition system contains the following components

- A Buffer β $w \in V$
- A Stack σ
- A Set of Arcs A

Models for DS: Transition Systems

- A Transition system contains the following components

- A Buffer β $w \in V$
- A Stack σ $w \in V$
- A Set of Arcs A

Models for DS: Transition Systems

- A Transition system contains the following components

● A Buffer	β	$w \in V$
● A Stack	σ	$w \in V$
● A Set of Arcs	A	$\subseteq V \times R \times V$

Models for DS: Transition Systems

- A Transition system contains the following components

- A Buffer β $w \in V$
- A Stack σ $w \in V$
- A Set of Arcs $A \subseteq V \times R \times V$

Configuration: Partial analysis of a sentence

Initial Configuration

β [economic news had little effect on financial markets .]

σ [root]

A []

Final Configuration

β []

σ [root]

A [

news	attribute	economic
had	subject	news
effect	attribute	little
markets	attribute	financial
had	subject	news
on	prep-obj	little
had	punct	.
had	predicate	root

]

Inference

- We wish to get from initial to final by applying (parameterized) actions:
 - Shift
 - Attach (left/right, label)
- How do we know which action to apply?

The Oracle

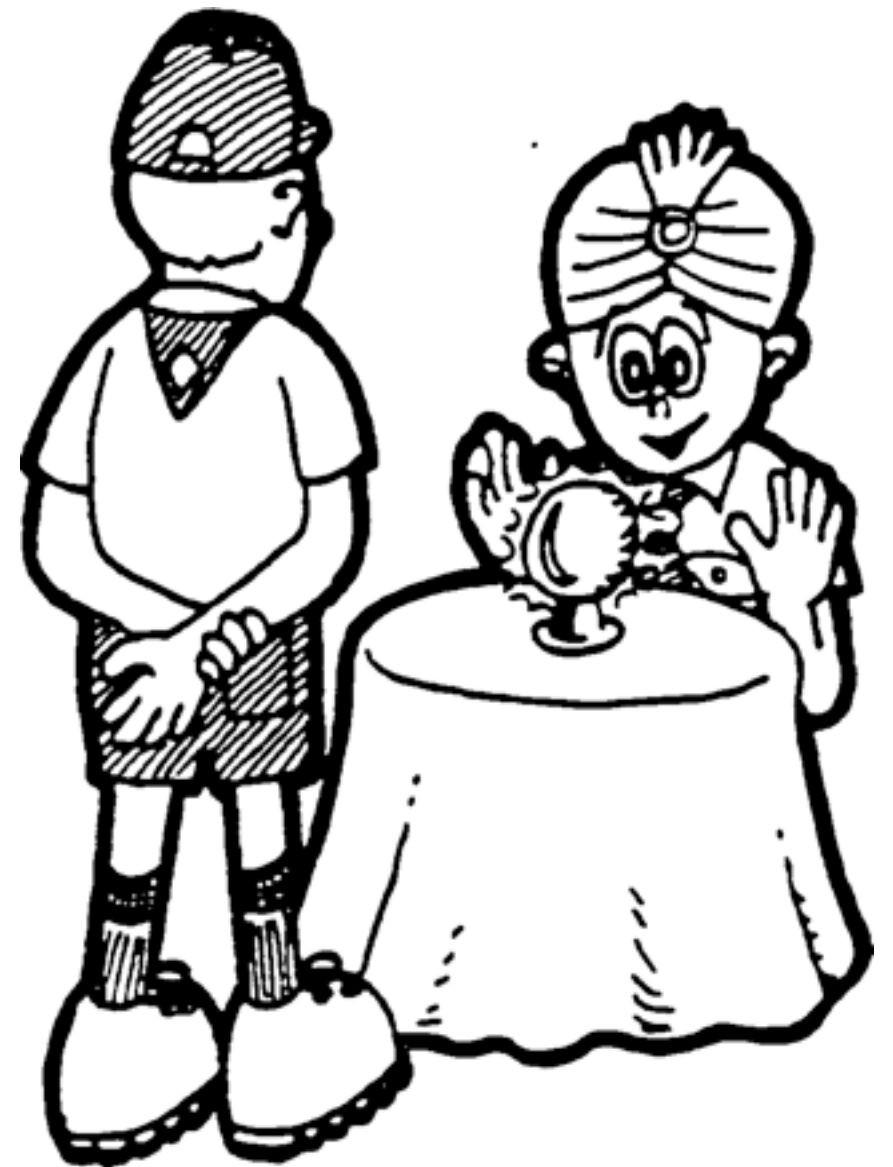
A Function from Configuration to Action

$$O : \mathcal{C} \rightarrow \mathcal{A}$$

The Oracle

A Function from Configuration to Action

$$O : \mathcal{C} \rightarrow \mathcal{A}$$



The Oracle

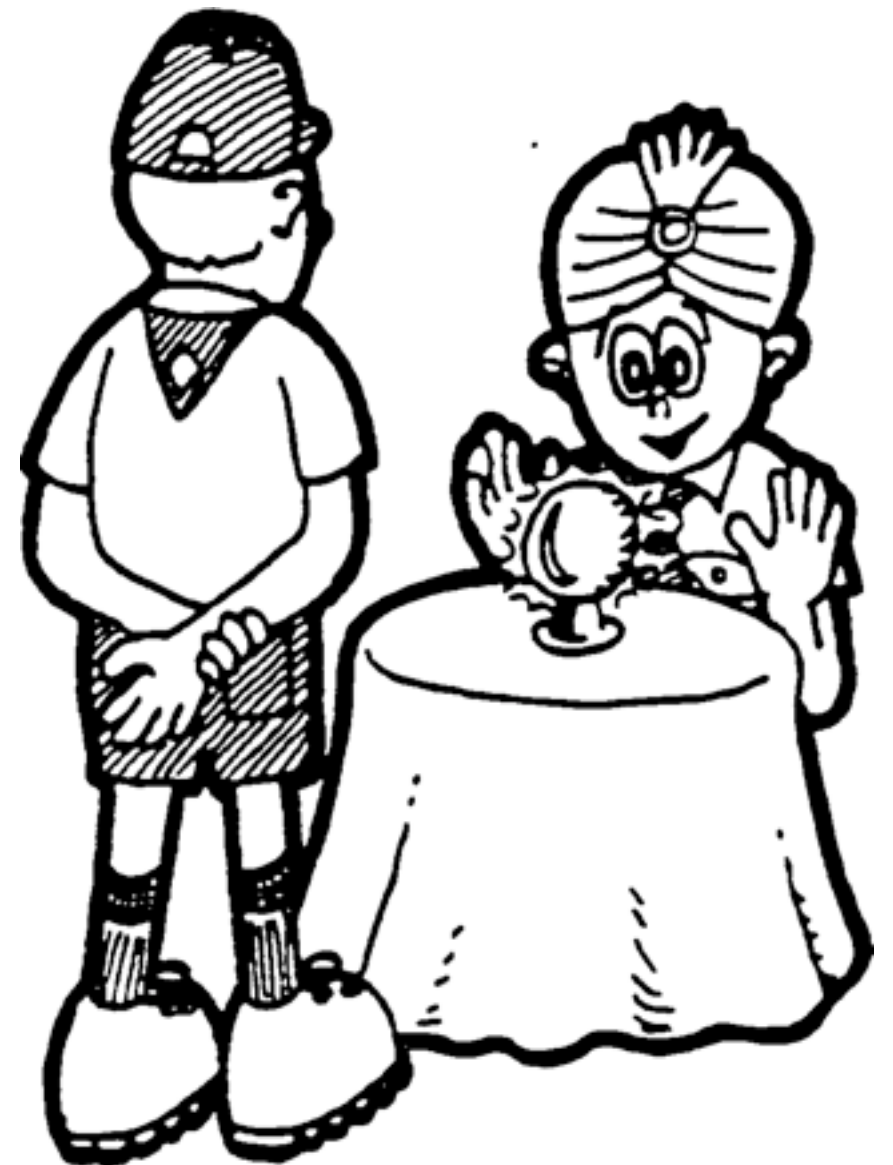
A Function from Configuration to Action

$$O : \mathcal{C} \rightarrow \mathcal{A}$$

SHIFT

ATTACH_Left

ATTACH_Right



Example:

Shift

β [economic news had little effect on financial markets .]

σ [root]

Example:

Shift

β [news had little effect on financial markets .]

σ [root, economic]

Example:

Shift

β [had little effect on financial markets .]

σ [root, economic, news]

Example:

Attach

β [had little effect on financial markets .]

σ [root, news]

news
|
economic

Example:

Shift

β [little effect on financial markets .]

σ [root, news, had]

news
|
economic

Example:

Attach

β [little effect on financial markets .]

σ [root, had]

had
|
news
|
economic

Example:

Shift

β [effect on financial markets .]

σ [root, had, little]

had
|
news
|
economic

Example:

Shift

β [on financial markets .]

σ [root, had, little, effect]

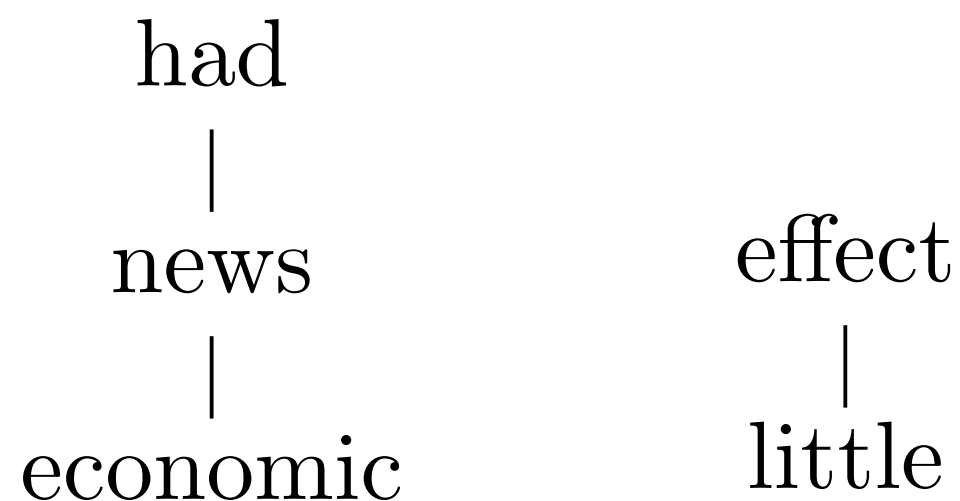
had
|
news
|
economic

Example:

Attach

β [on financial markets .]

σ [root, had, effect]



Example:

Shift
Shift
Shift

β [.]

σ [root, had, effect, on, financial, markets]

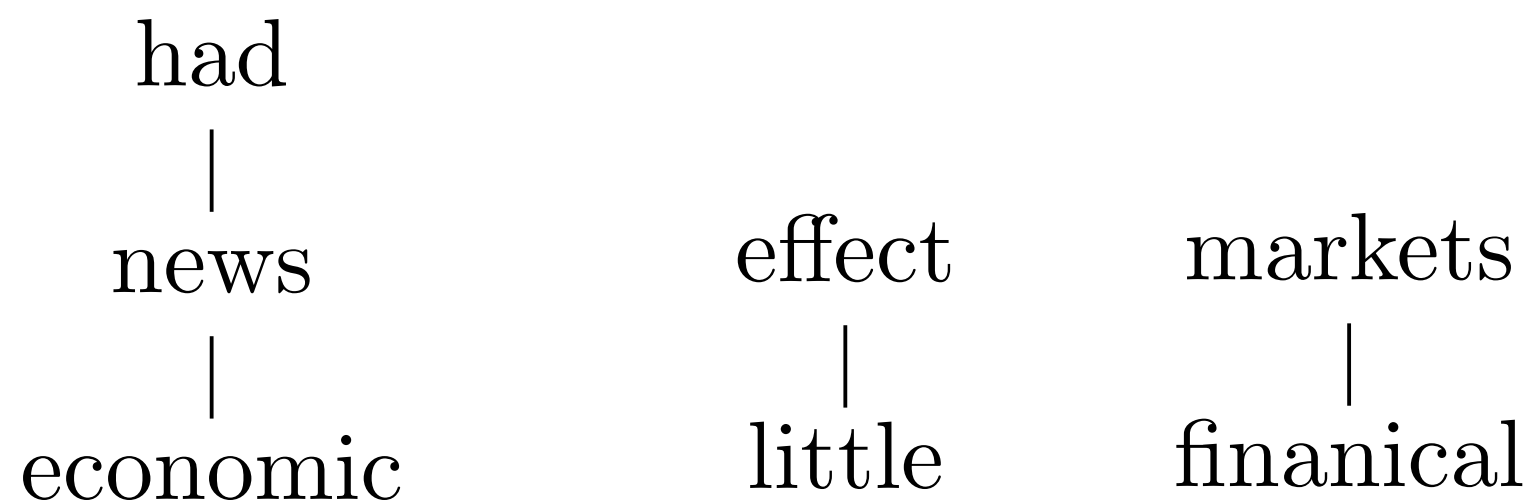
had	
news	effect
economic	little

Example:

Attach

β [.]

σ [root, had, effect, on, markets]



Example:

Attach

β [.]

σ [root, had, effect, on]

had
|
news
|
economic

effect
|
little

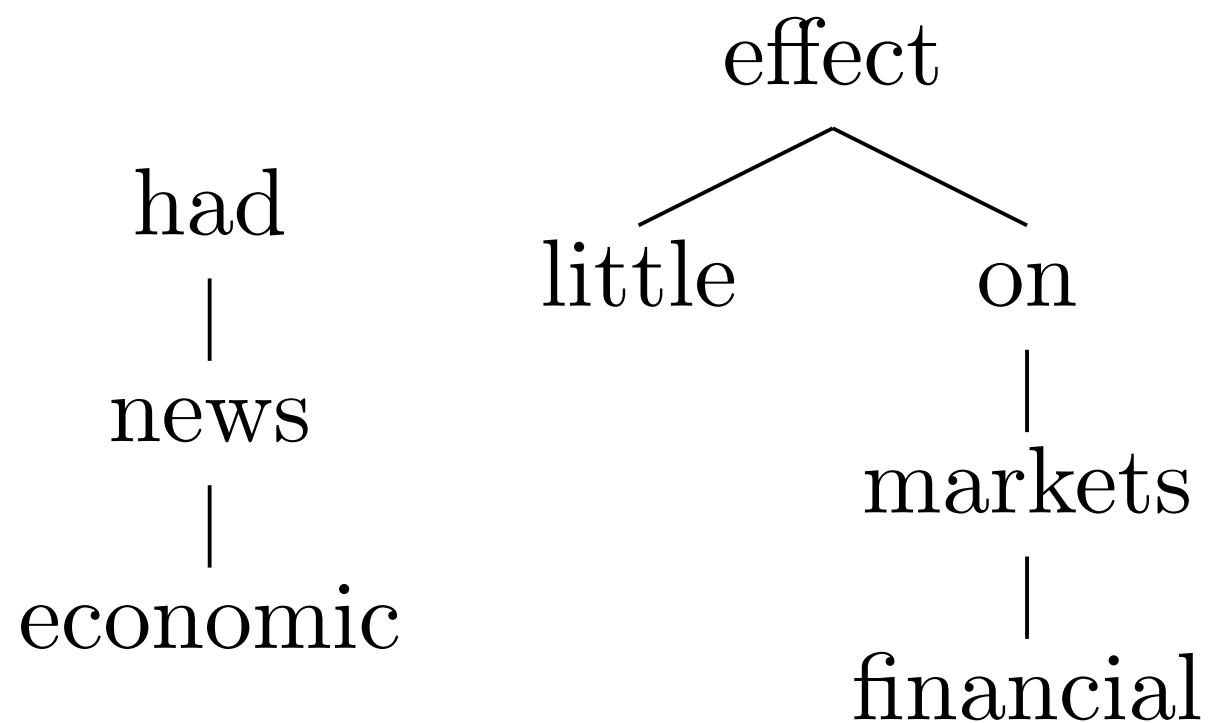
on
|
markets
|
financial

Example:

Attach

β [.]

σ [root, had, effect]

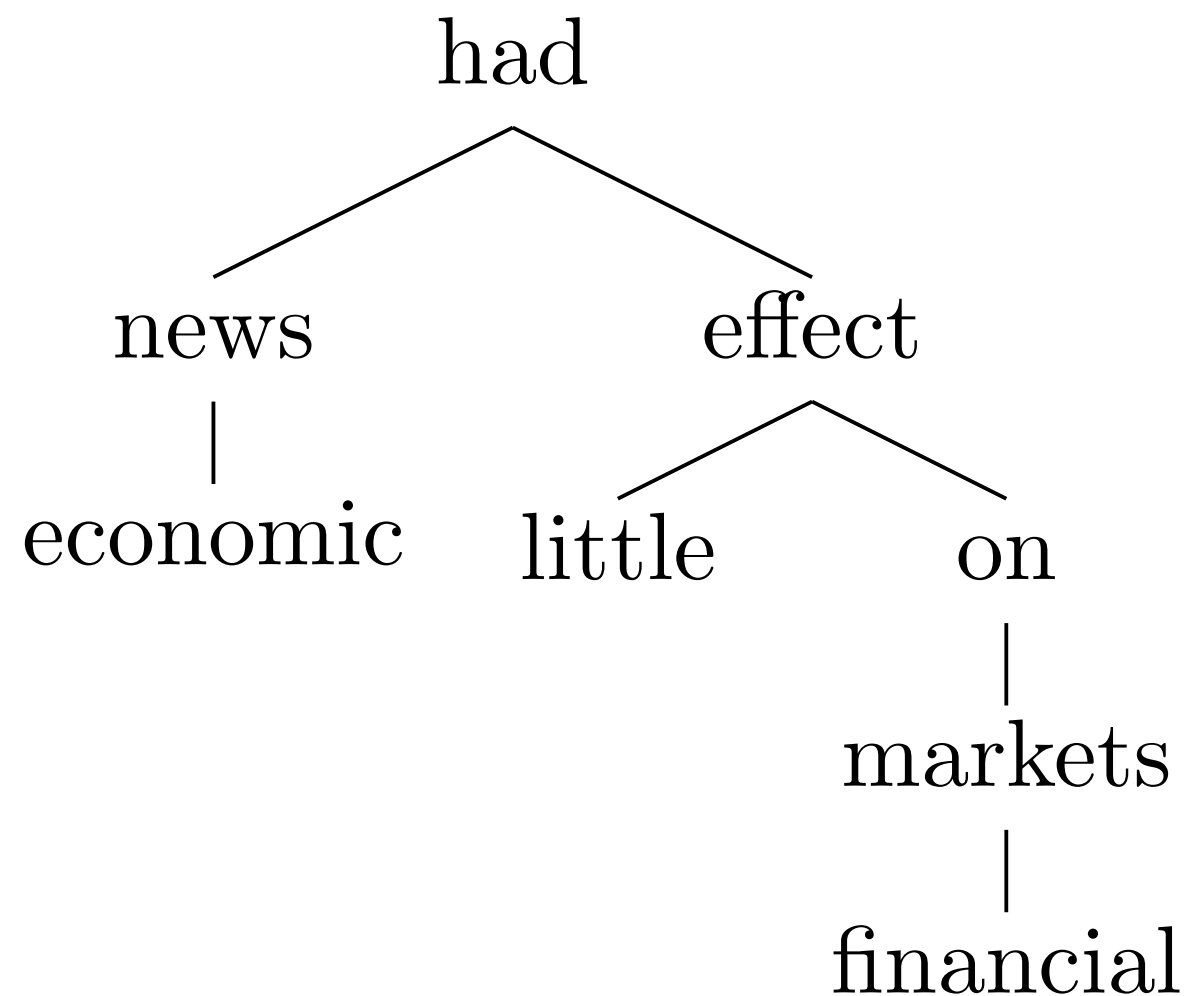


Example:

Attach

β [.]

σ [root, had]

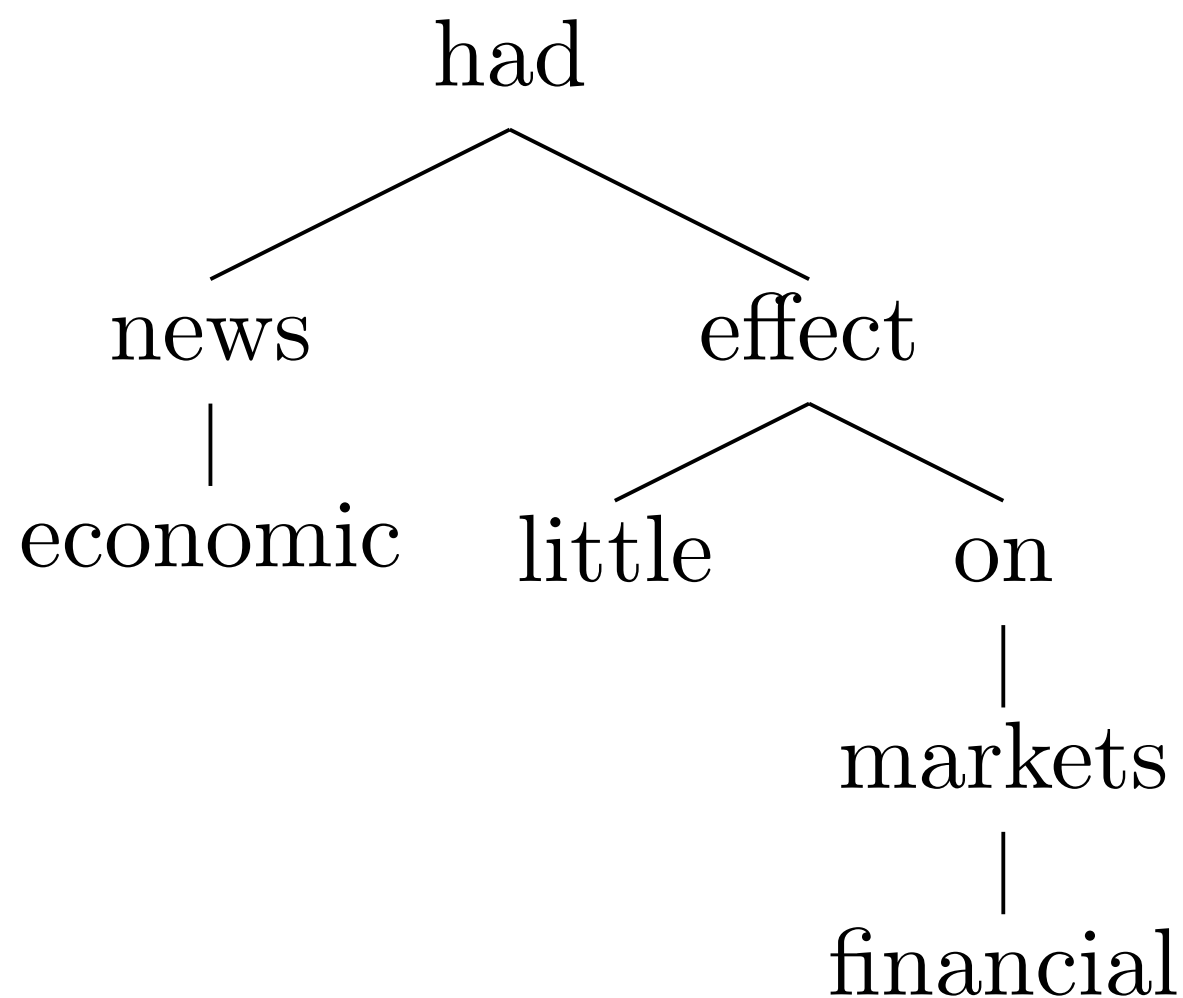


Example:

Shift

β []

σ [root, had,.]

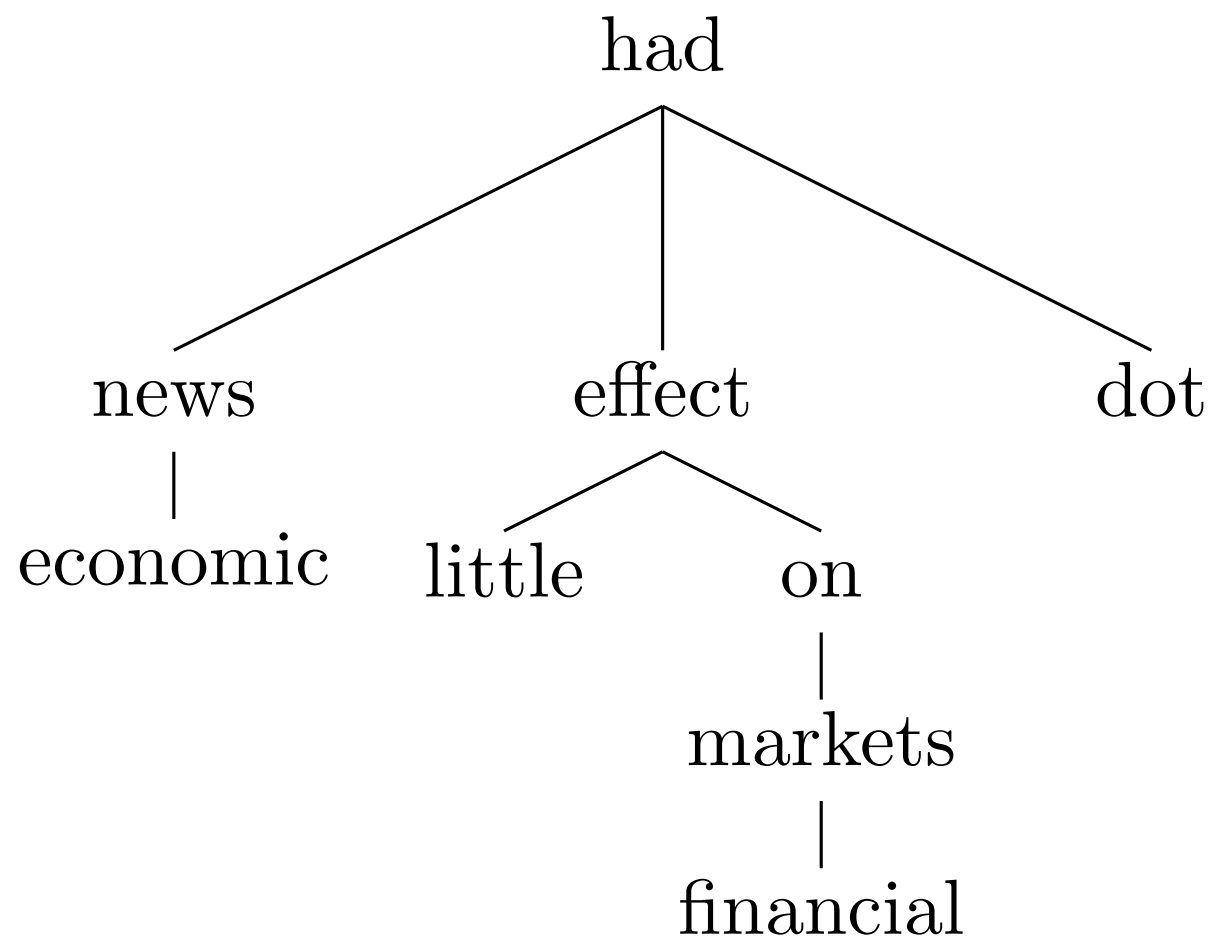


Example:

Attach

β []

σ [root, had]

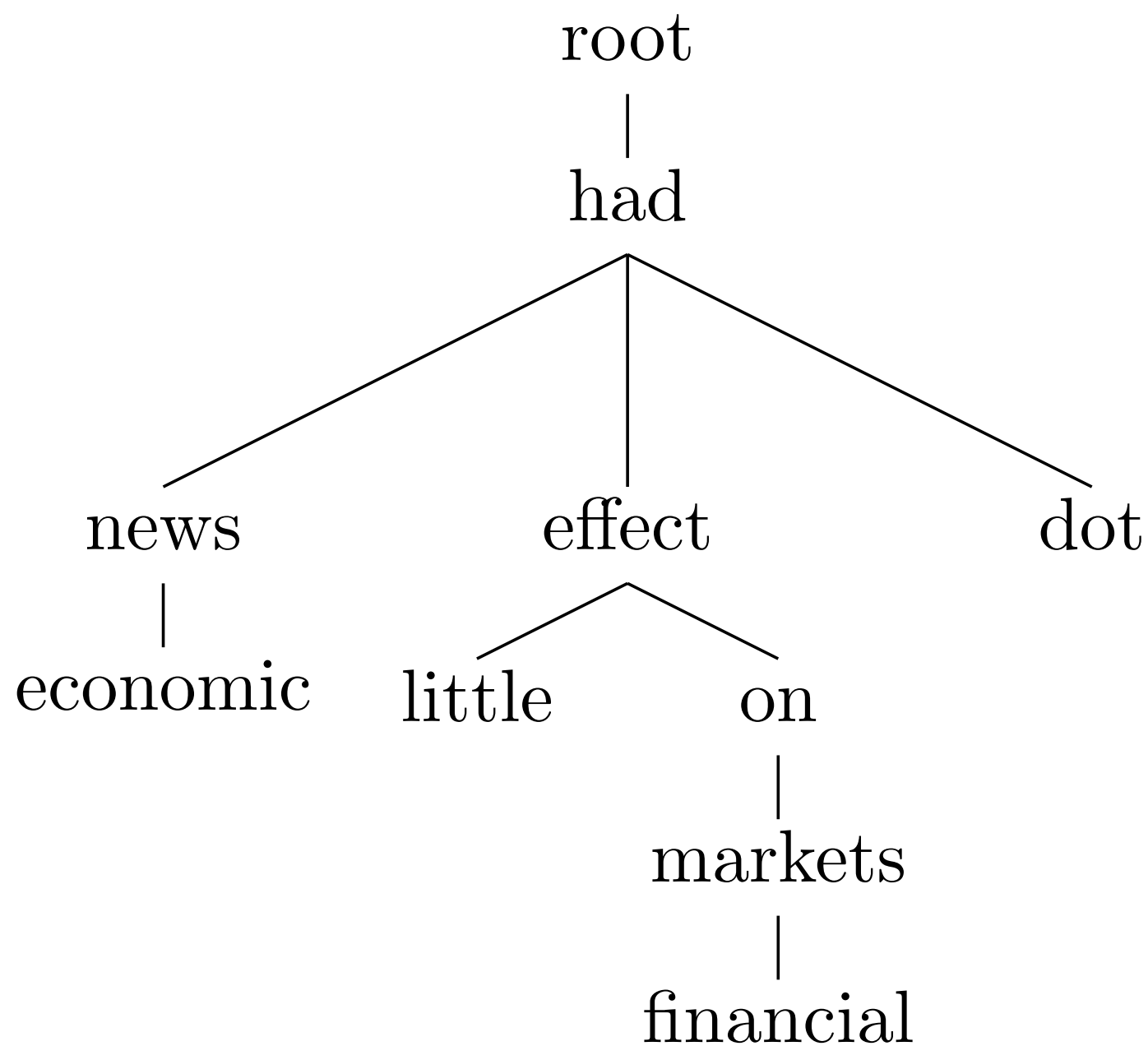


Example:

Attach

β []

σ [root]



The Oracle

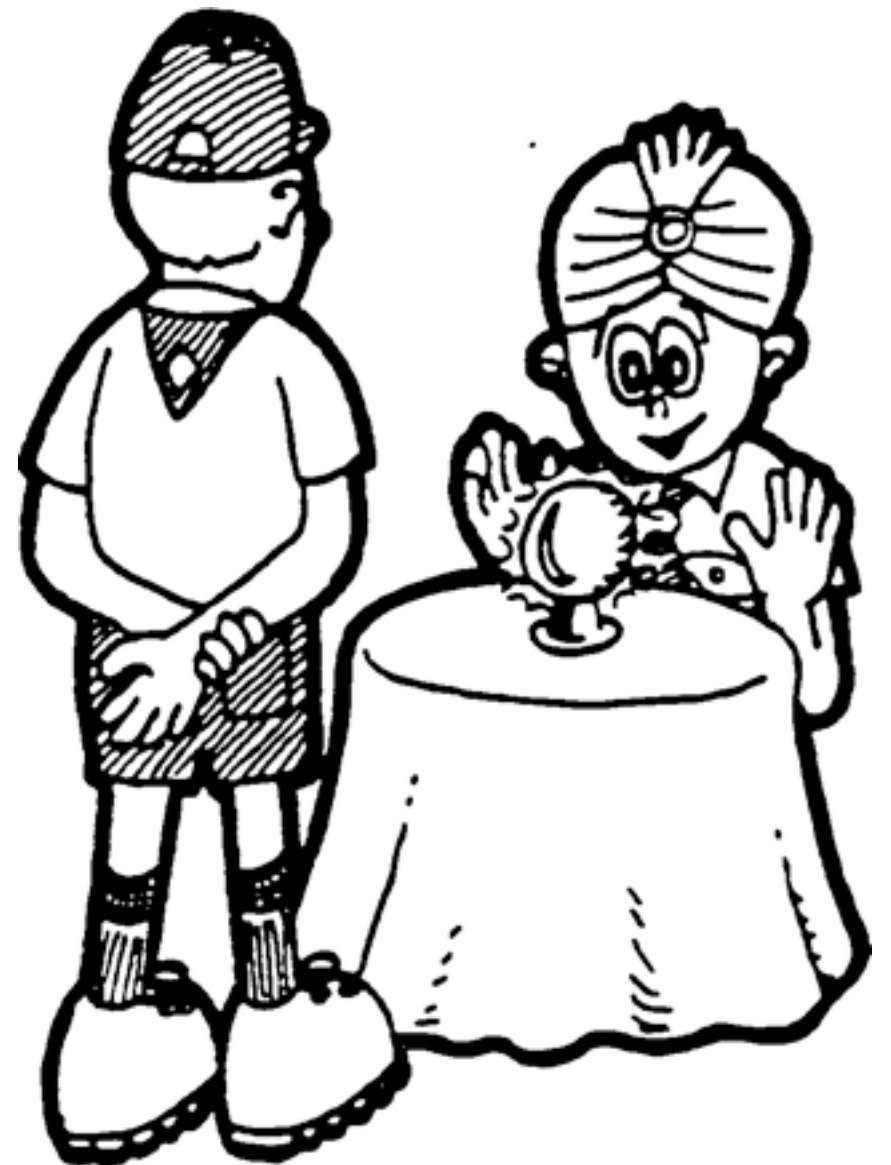
$$O : \mathcal{C} \rightarrow \mathcal{A}$$

SHIFT

ATTACH_Left

ATTACH_Right

$$\langle c_0, c_1, c_2, \dots, c_n, c_f \rangle$$



The Oracle

Arc-Standard

$$O : \mathcal{C} \rightarrow \mathcal{A}$$

Shift $(\sigma, w_i | \beta, A) \Rightarrow (\sigma | w_i, \beta, A)$

Left_Arc $(\sigma | w_i, w_j | \beta, A) \Rightarrow (\sigma, w_j | \beta, A \cup (w_j, r, w_i))$

Right_Arc $(\sigma | w_i, w_j | \beta, A) \Rightarrow (\sigma, w_i | \beta, A \cup (w_i, r, w_j))$

$\langle c_0, c_1, c_2, \dots, c_n, c_f \rangle \longleftarrow$ Transition sequence

Example:

Arc-Standard

β [economic news had little effect on financial markets .]

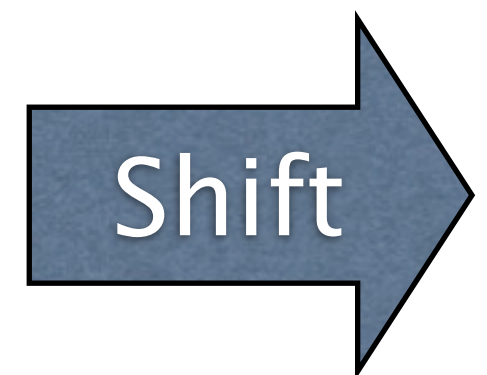
σ [root]

Example:

Arc-Standard

β [economic news had little effect on financial markets .]

σ [root]



Example:

Arc-Standard

β [news had little effect on financial markets .]

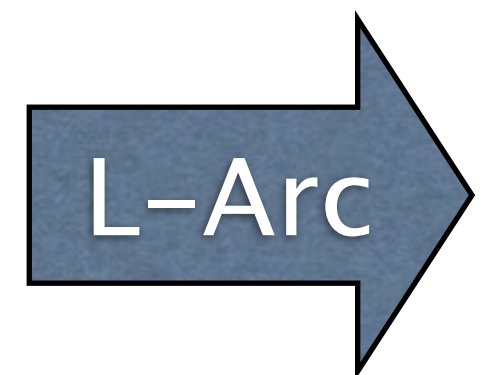
σ [root, economic]

Example:

Arc-Standard

β [news had little effect on financial markets .]

σ [root, economic]



Example:

Arc-Standard

β [news had little effect on financial markets .]

σ [root]

news
|
economic

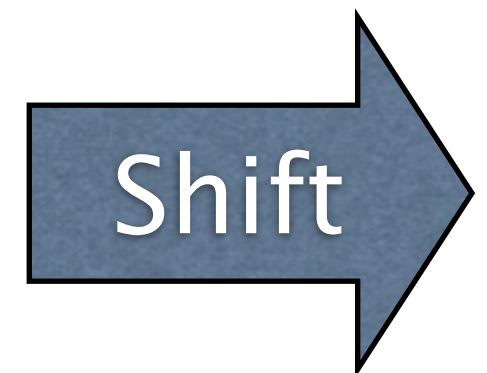
Example:

Arc-Standard

β [news had little effect on financial markets .]

σ [root]

news
|
economic



Example:

Arc-Standard

β [had little effect on financial markets .]

σ [root, news]

news
|
economic

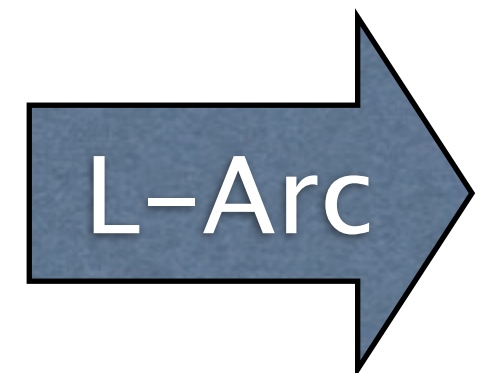
Example:

Arc-Standard

β [had little effect on financial markets .]

σ [root, news]

news
|
economic



Example:

Arc-Standard

β [had, little effect on financial markets .]

σ [root]

had
|
news
|
economic

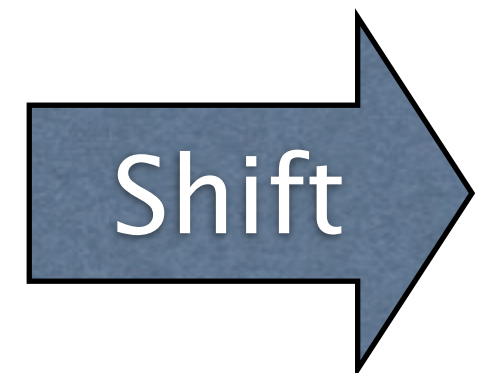
Example:

Arc-Standard

β [had, little effect on financial markets .]

σ [root]

had
|
news
|
economic



Example:

Arc-Standard

β [little effect on financial markets .]

σ [root, had]

had
|
news
|
economic

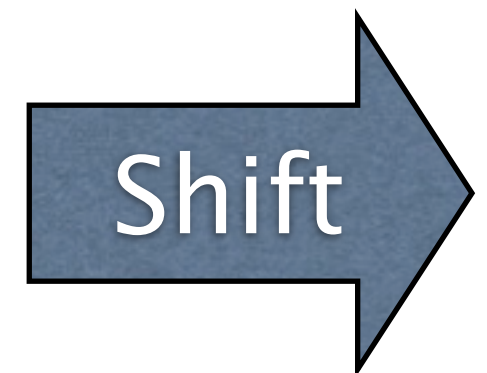
Example:

Arc-Standard

β [little effect on financial markets .]

σ [root, had]

had
|
news
|
economic



Example:

Arc-Standard

β [effect on financial markets .]

σ [root, had, little]

had
|
news
|
economic

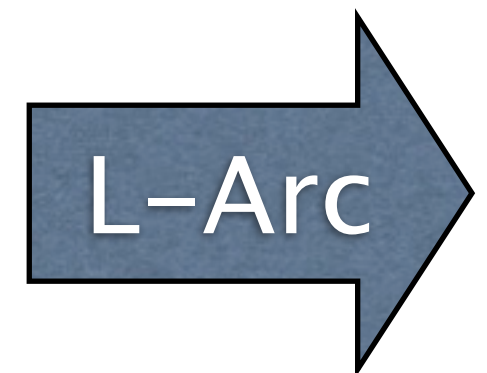
Example:

Arc-Standard

β [effect on financial markets .]

σ [root, had, little]

had
|
news
|
economic



Example:

Arc-Standard

β [effect, on financial markets .]

σ [root, had]



Example:

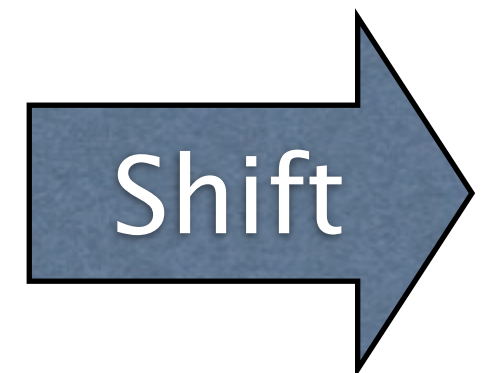
Arc-Standard

β [effect, on financial markets .]

σ [root, had]

had
|
news
|
economic

effect
|
little

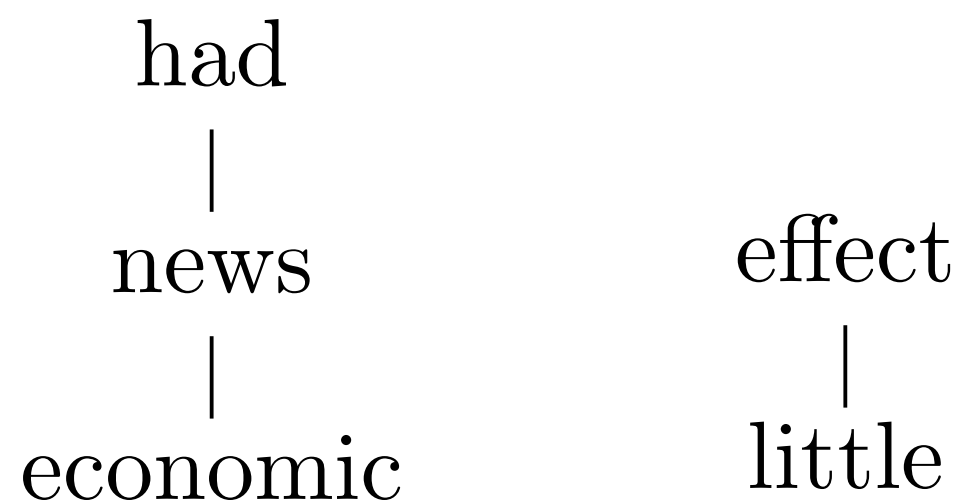


Example:

Arc-Standard

β [on financial markets .]

σ [root, had, effect]



Example:

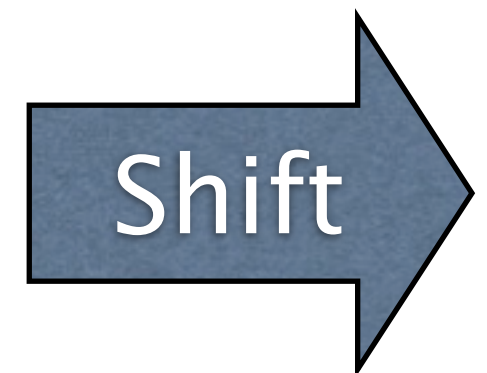
Arc-Standard

β [on financial markets .]

σ [root, had, effect]

had
|
news
|
economic

effect
|
little

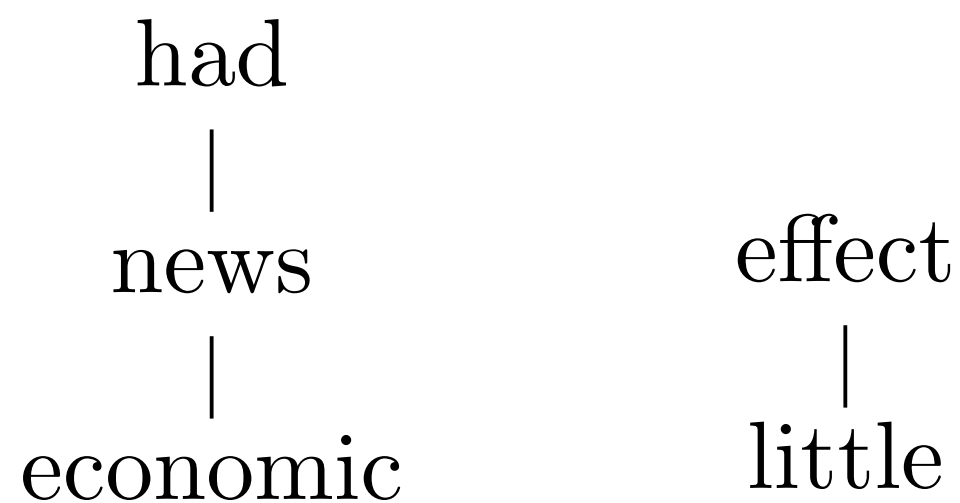


Example:

Arc-Standard

β [financial markets .]

σ [root, had, effect, on]



Example:

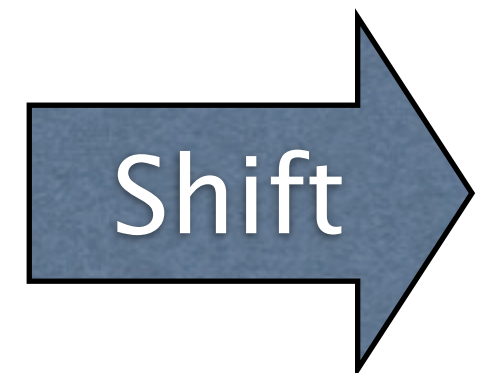
Arc-Standard

β [financial markets .]

σ [root, had, effect, on]

had
|
news
|
economic

effect
|
little



Example:

Arc-Standard

β [markets .]

σ [root, had, effect, on, financial]



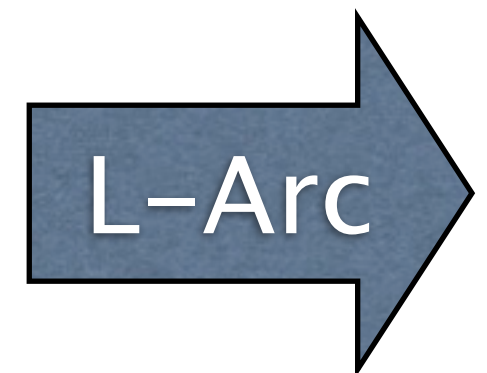
Example:

Arc-Standard

β [markets .]

σ [root, had, effect, on, financial]

had	
news	effect
economic	little

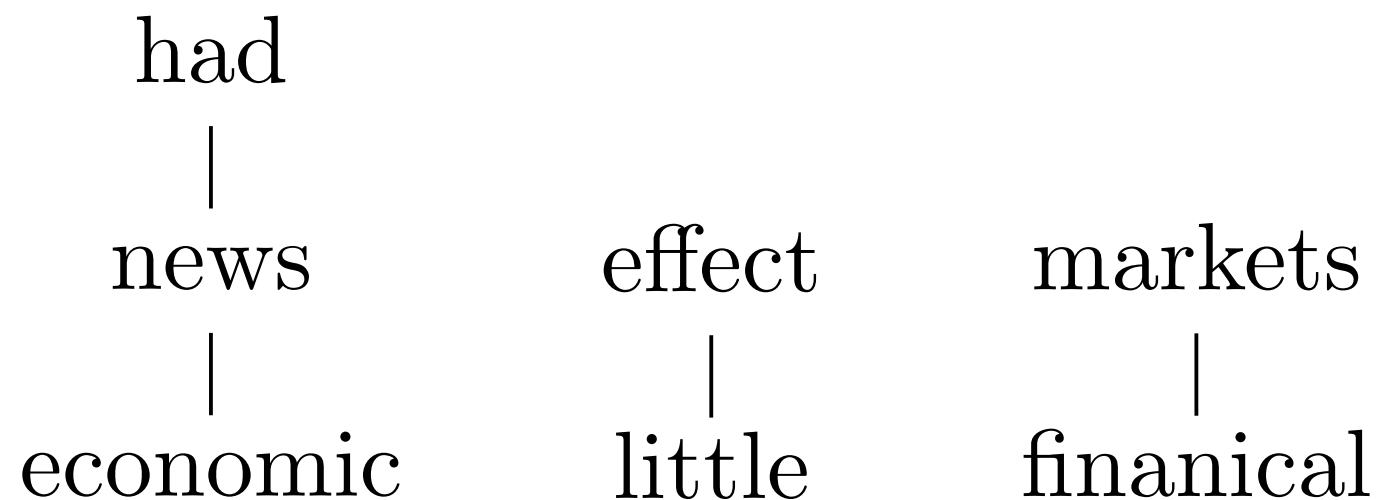


Example:

Arc-Standard

β [markets .]

σ [root, had, effect, on]

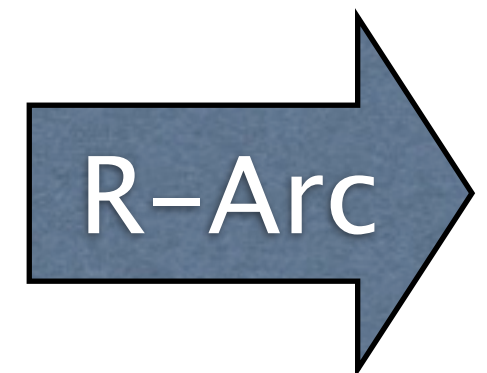
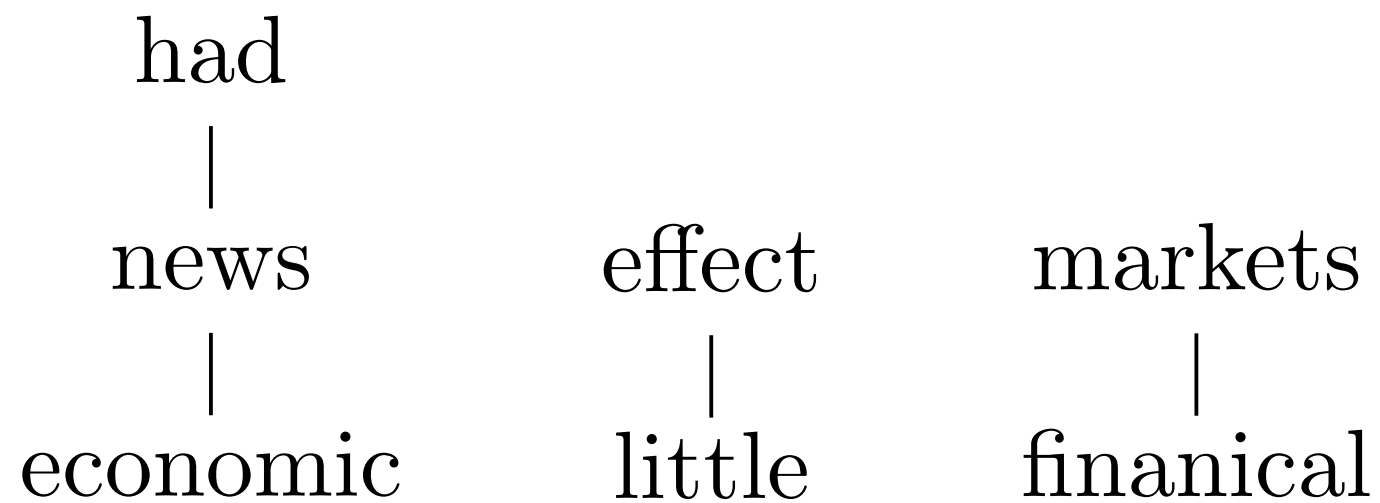


Example:

Arc-Standard

β [markets .]

σ [root, had, effect, on]



Example:

Arc-Standard

β [on .]

σ [root, had, effect]



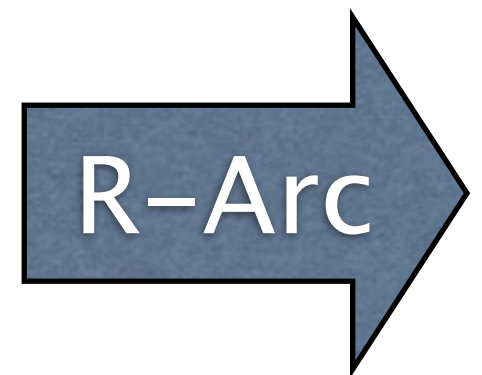
Example:

Arc-Standard

β [on .]

σ [root, had, effect]

had		on
news	effect	markets
economic	little	financial

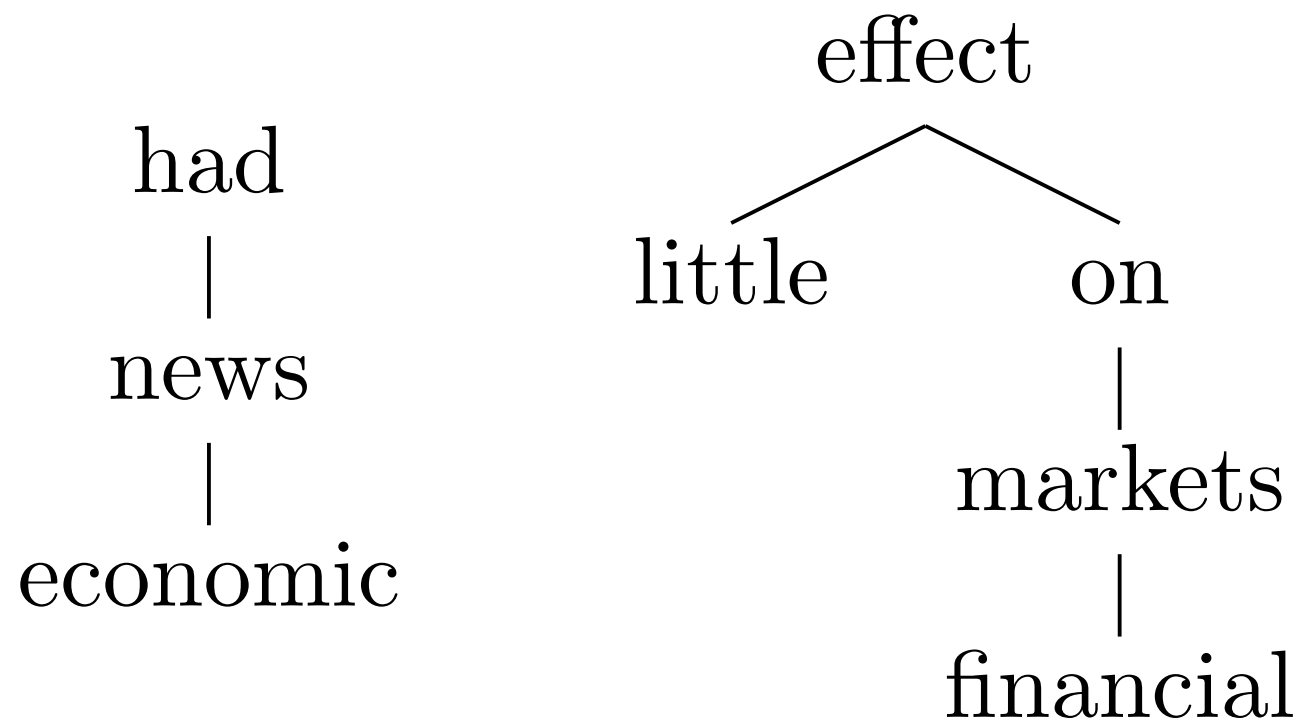


Example:

Arc-Standard

β [effect .]

σ [root, had]

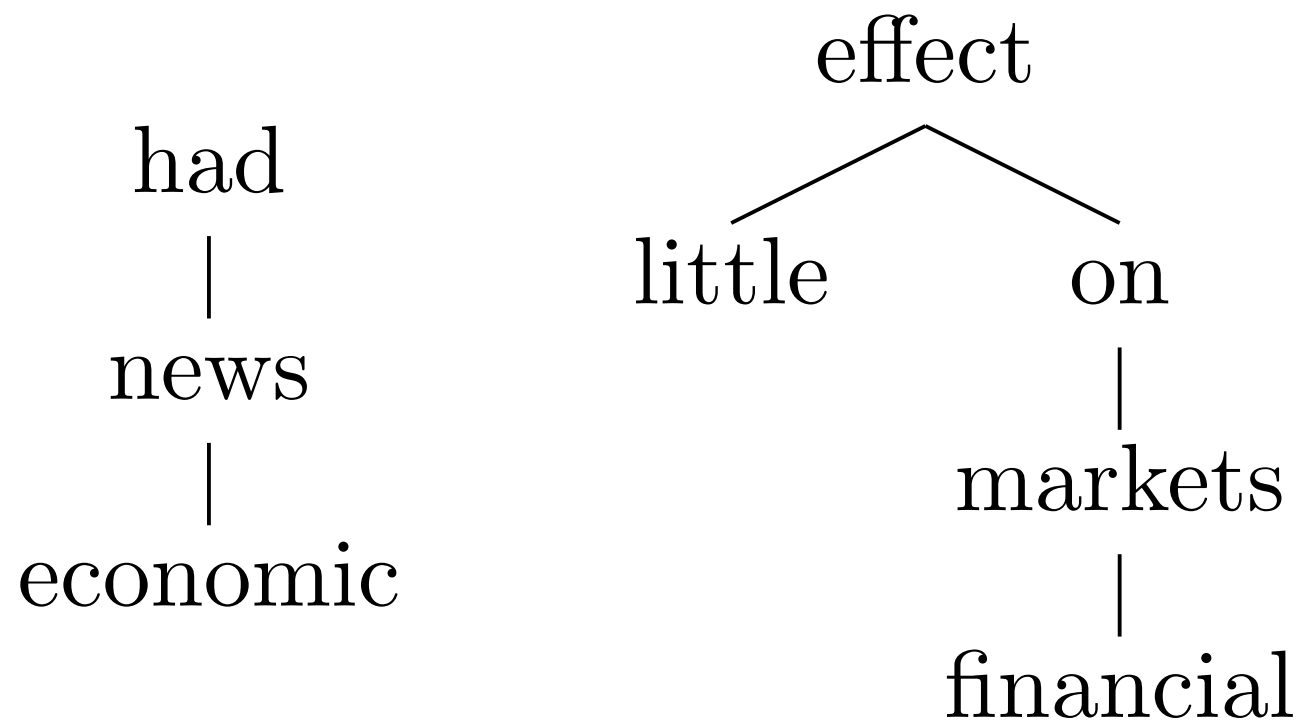


Example:

Arc-Standard

β [effect .]

σ [root, had]



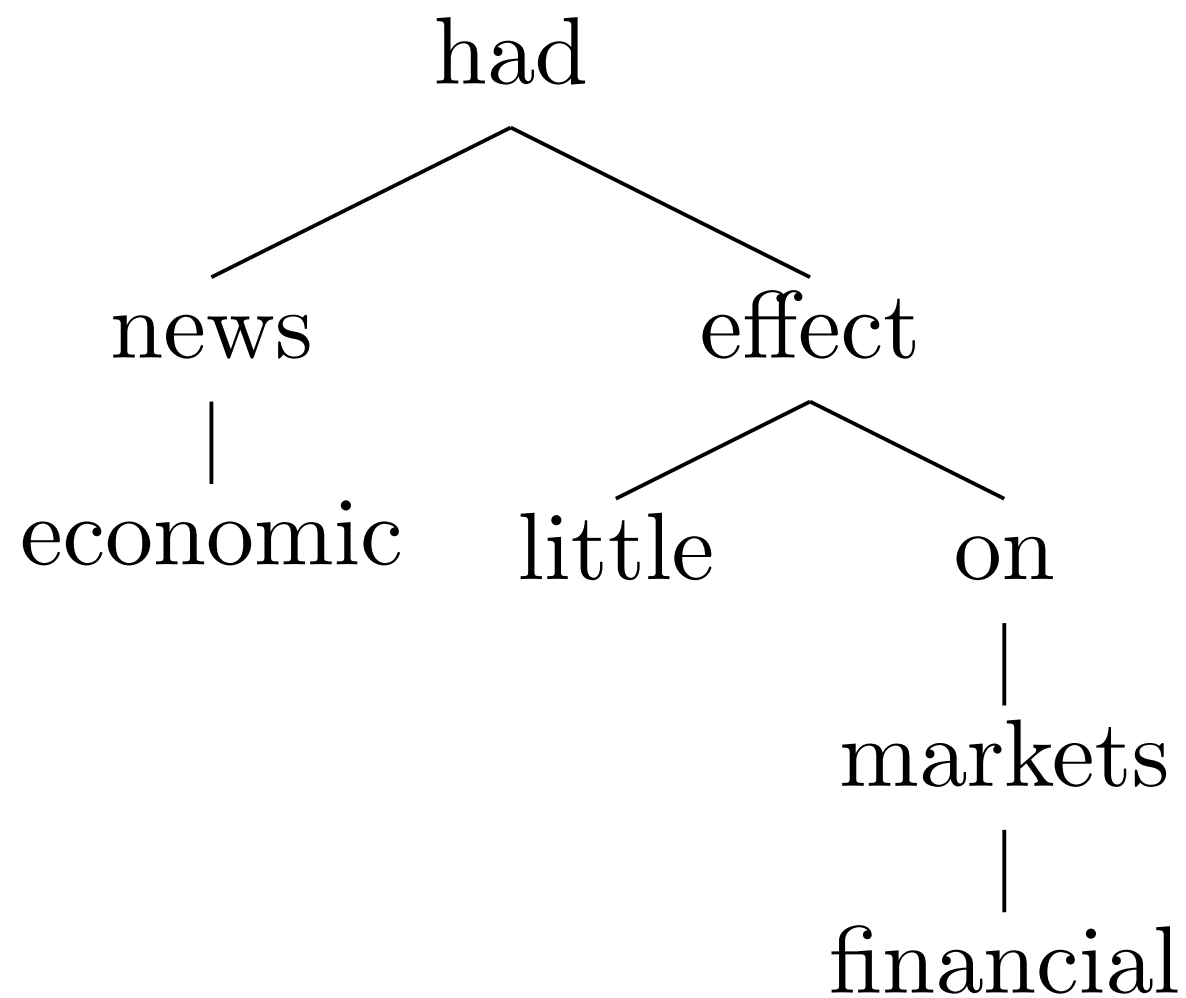
R-Arc

Example:

Arc-Standard

β [had .]

σ [root]

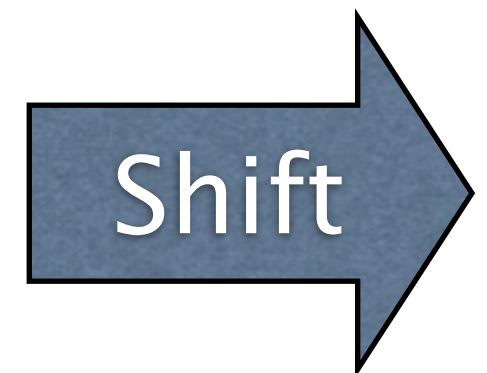
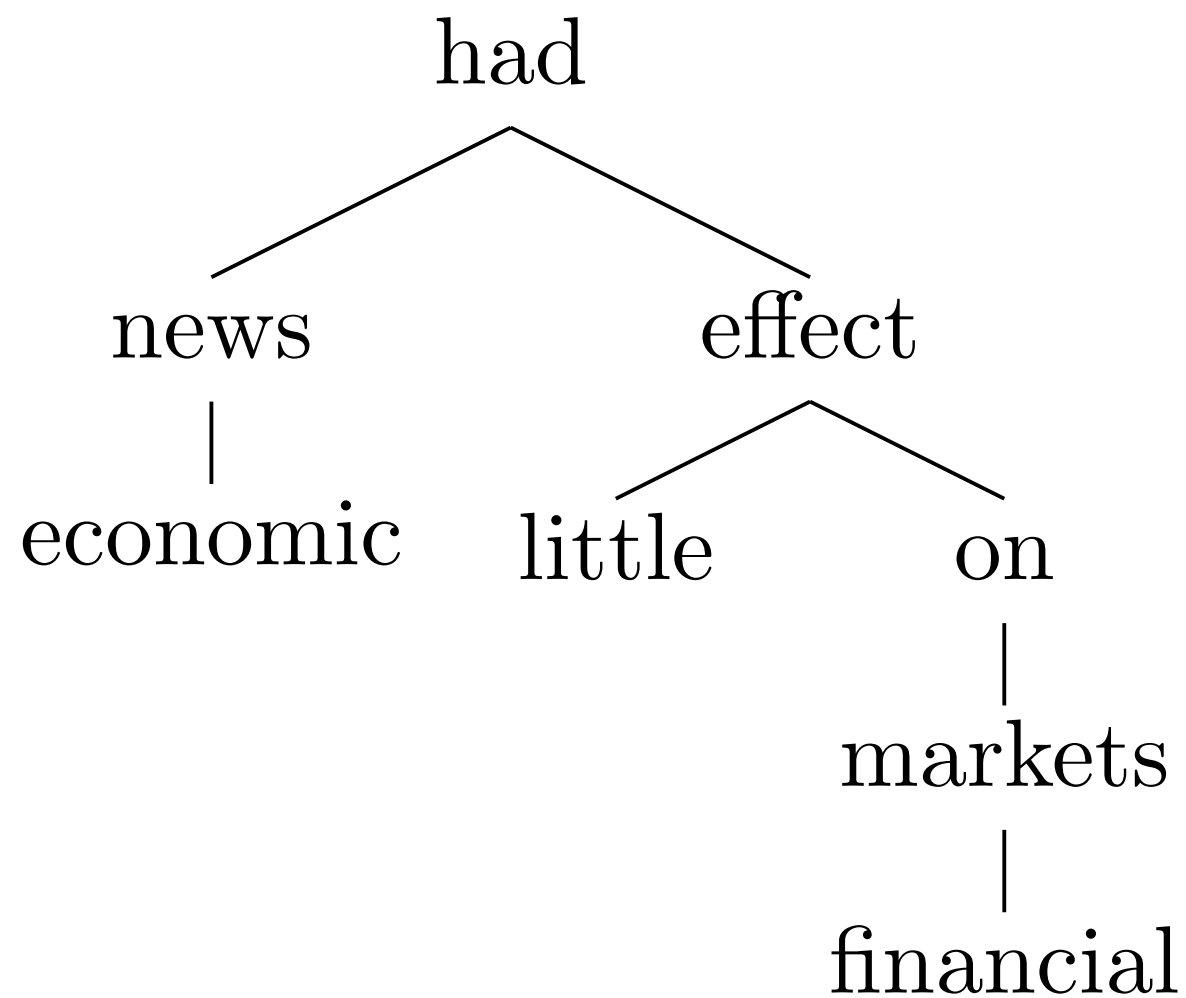


Example:

Arc-Standard

β [had .]

σ [root]

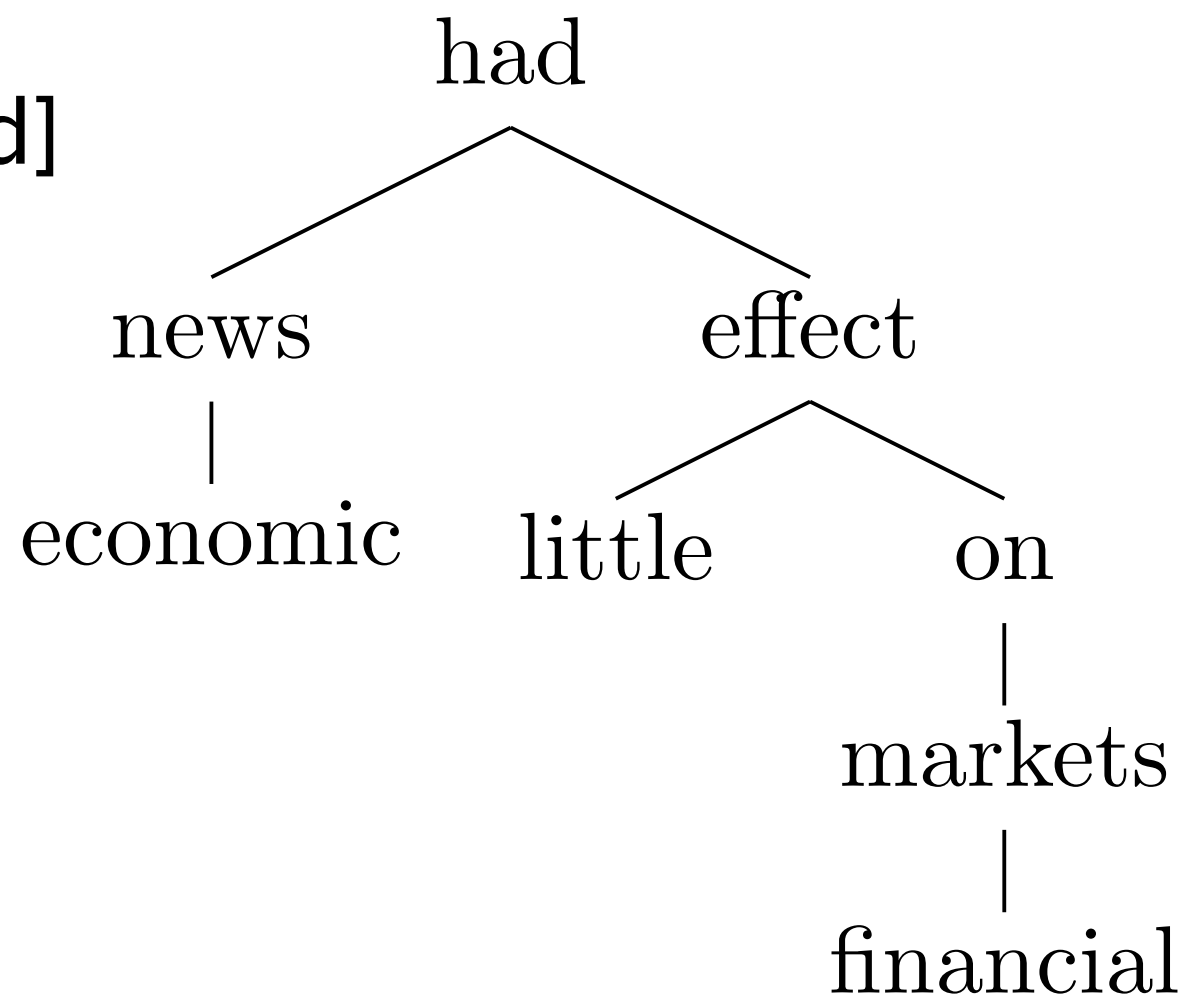


Example:

Arc-Standard

β [.]

σ [root, had]

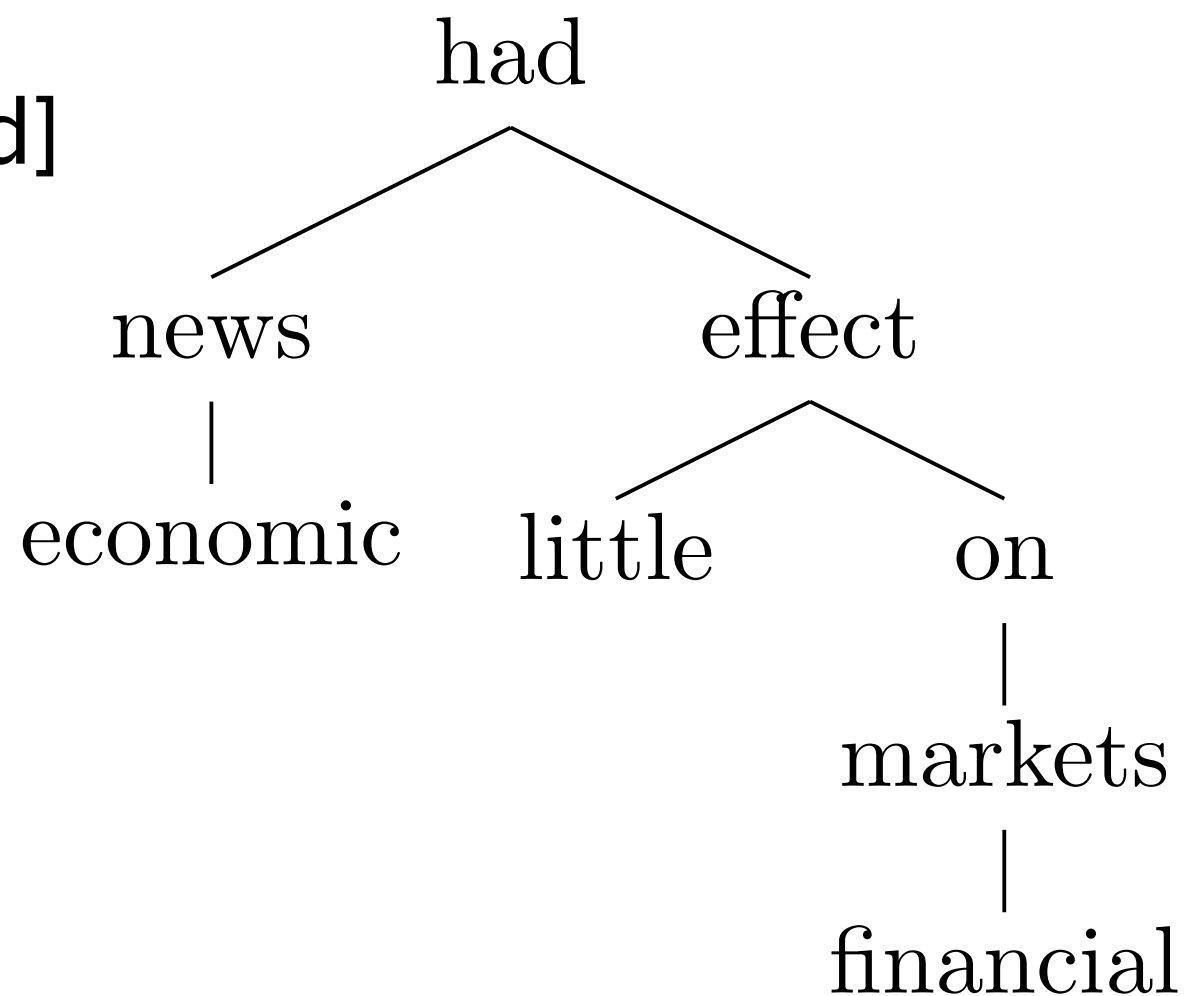


Example:

Arc-Standard

β [.]

σ [root, had]



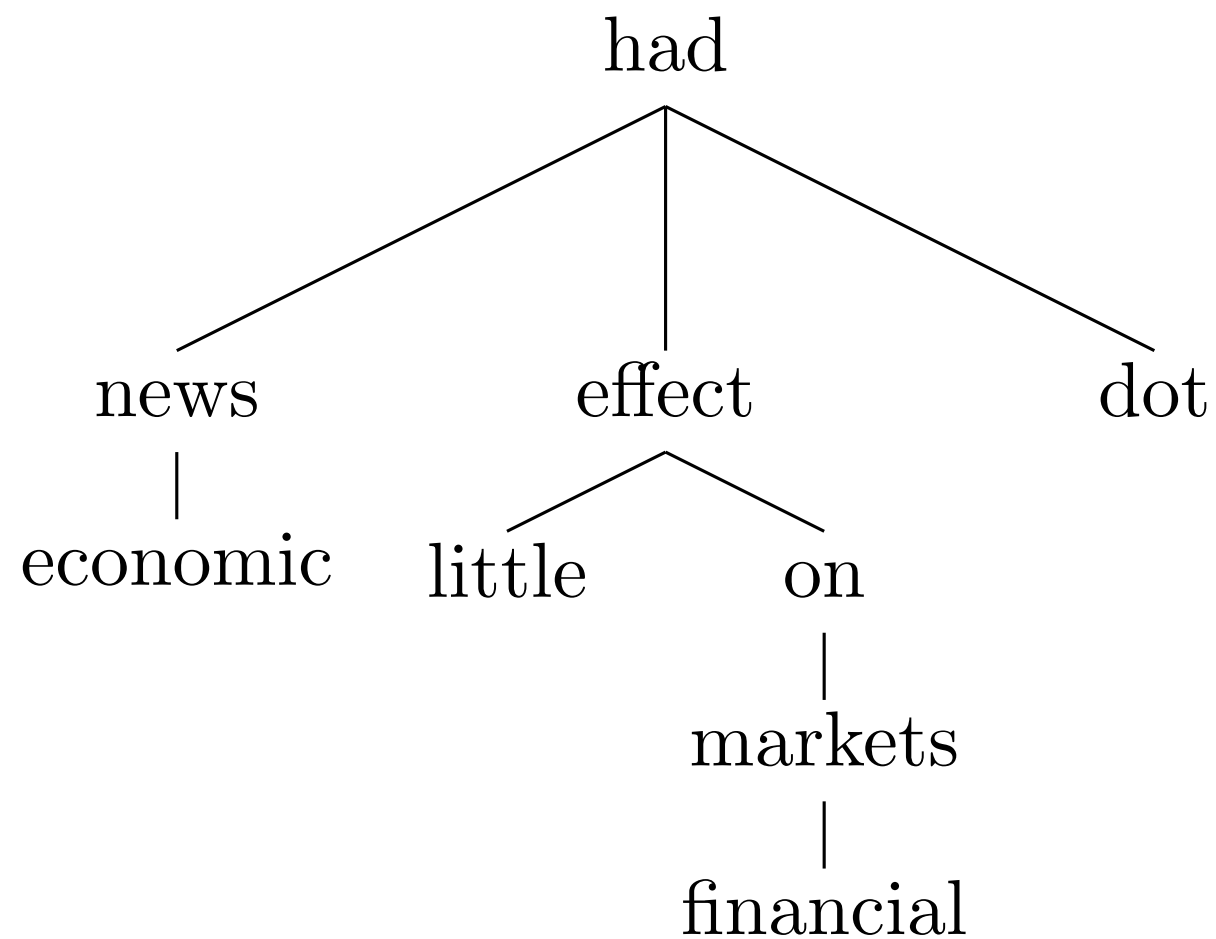
R-Arc

Example:

Arc-Standard

β [had]

σ [root]

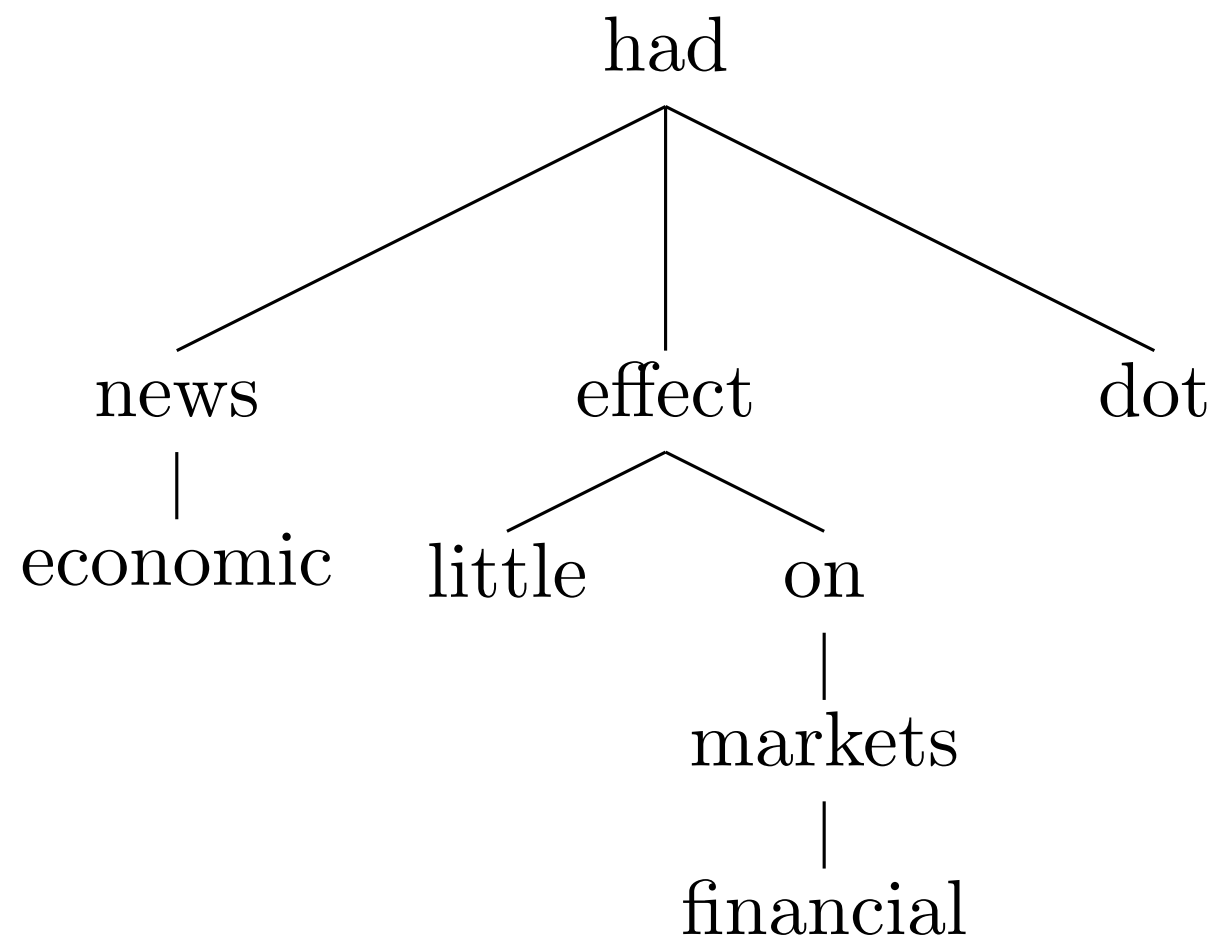


Example:

Arc-Standard

β [had]

σ [root]



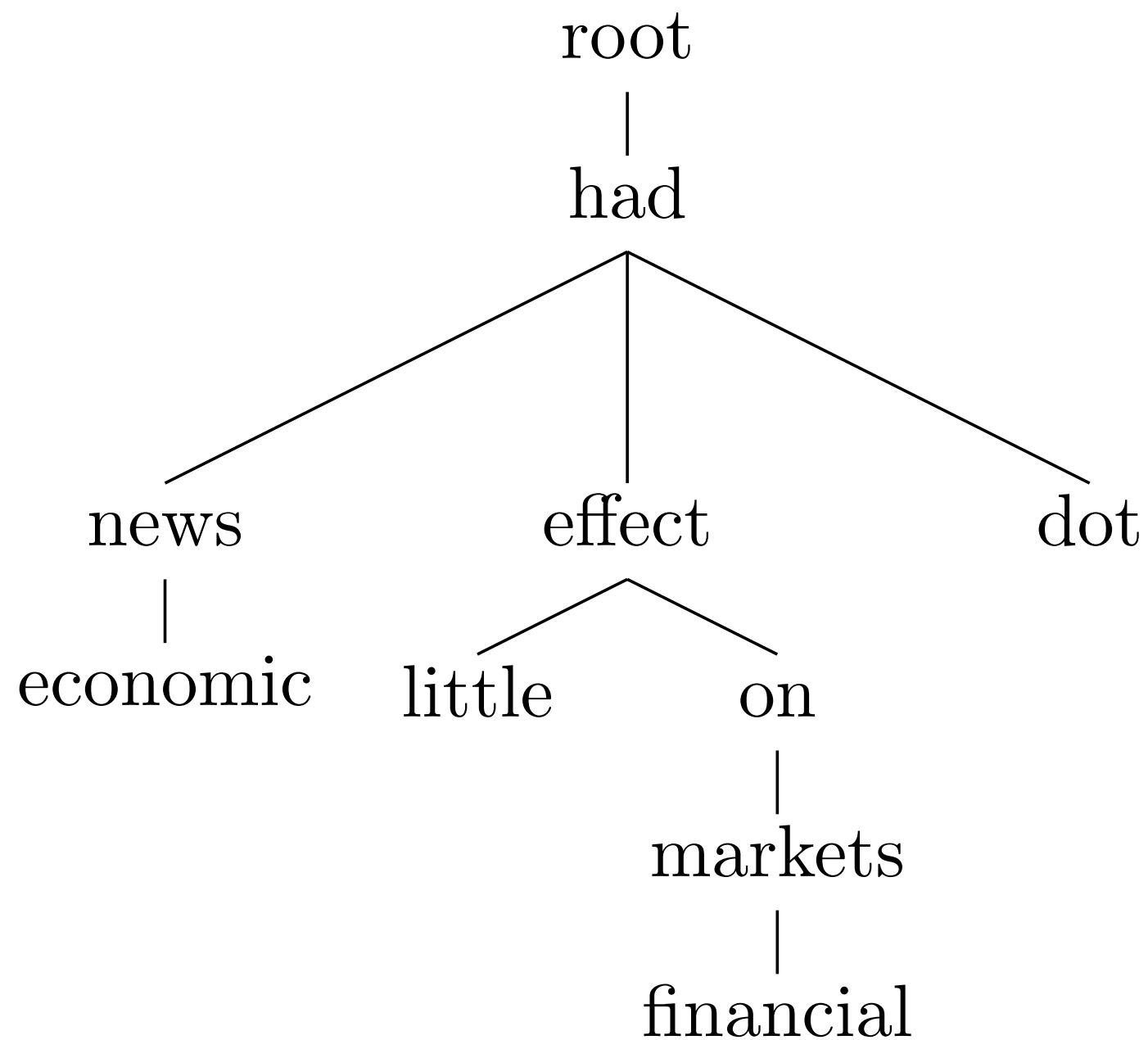
R-Arc

Example:

Arc-Standard

β [root]

σ []

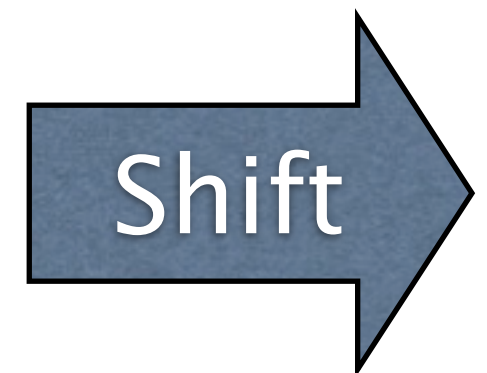
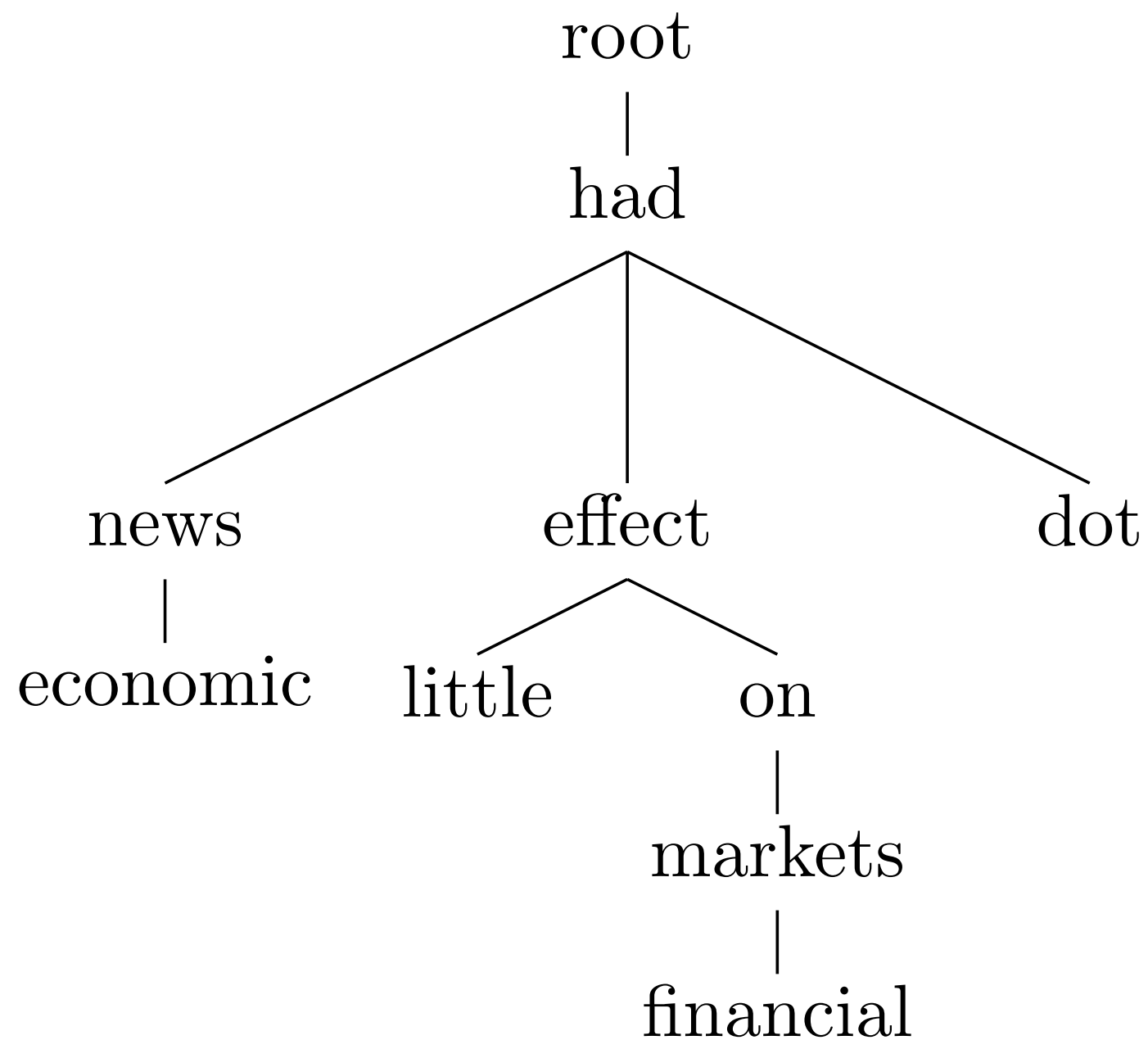


Example:

Arc-Standard

β [root]

σ []

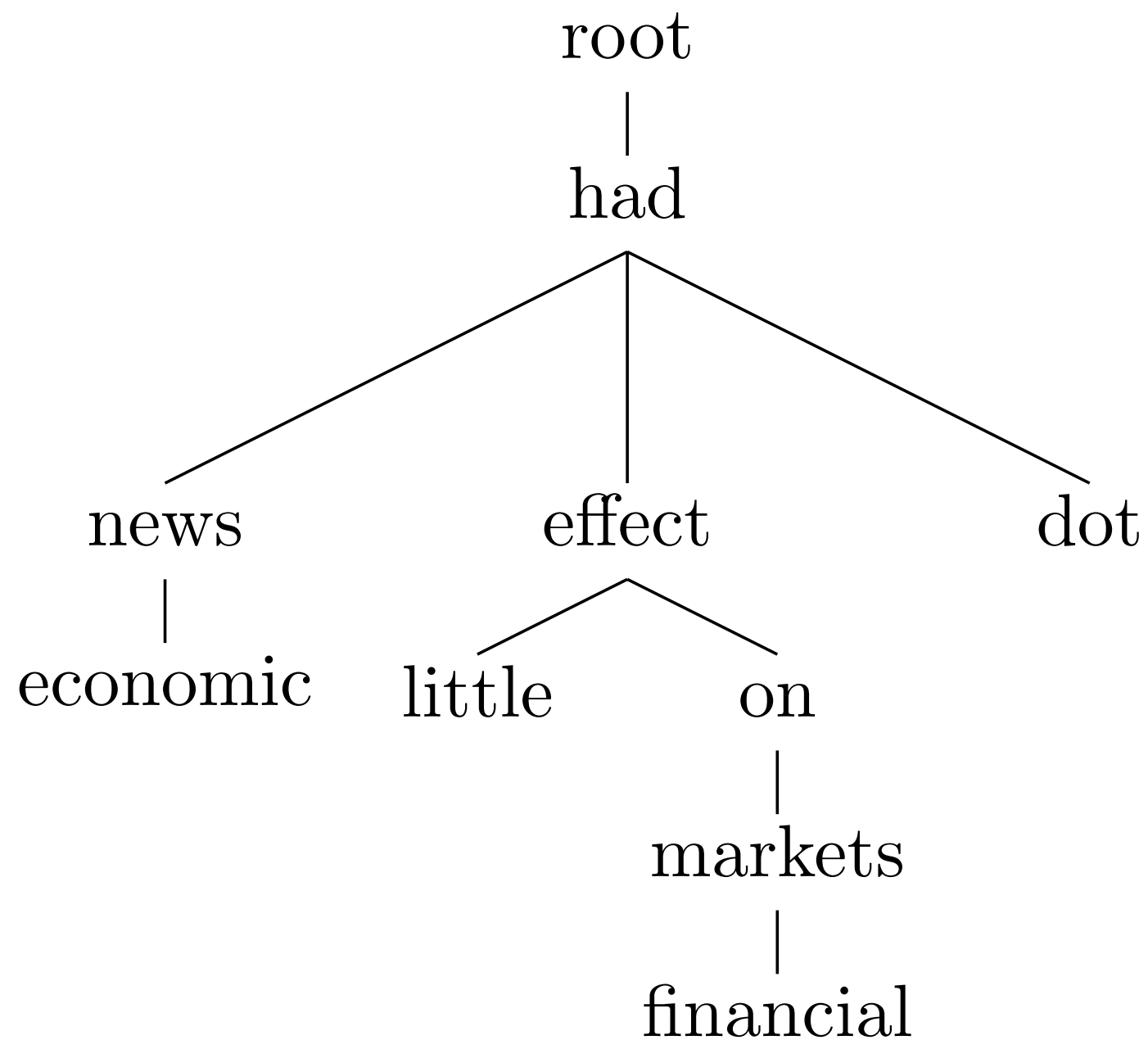


Example:

Arc-Standard

β []

σ [root]



Done

The Oracle

Arc-Standard

$$O : \mathcal{C} \rightarrow \mathcal{A}$$

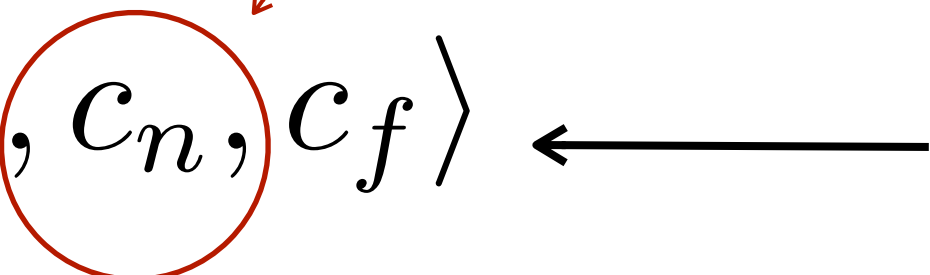
SHIFT

$$(\sigma, w_i | \beta, A) \Rightarrow (\sigma | w_i, \beta, A)$$

ATTACH_Left $(\sigma | w_i, w_j | \beta, A) \Rightarrow (\sigma, w_i | \beta, A \cup (w_i, r, w_j))$

ATTACH_Right $(\sigma | w_i, w_j | \beta, A) \Rightarrow (\sigma, w_j | \beta, A \cup (w_j, r, w_i))$

$\langle c_0, c_1, c_2, \dots, c_n, c_f \rangle$ ← Transition sequence



2x|sentence|

The Oracle

Arc-Eager

$$O : \mathcal{C} \rightarrow \mathcal{A}$$

Shift $(\sigma, w_i | \beta, A) \Rightarrow (\sigma | w_i, \beta, A)$

Left_Arc $(\sigma | w_i, w_j | \beta, A) \Rightarrow (\sigma, w_j | \beta, A \cup \{(w_j, r, w_i)\})$

Right_Arc $(\sigma | w_i, w_j | \beta, A) \Rightarrow (\sigma | w_i | w_j, \beta, A \cup \{(w_i, r, w_j)\})$

Reduce $(\sigma | w_i, \beta, A) \Rightarrow (\sigma, \beta, A)$

Example:

Arc-Eager

β [economic news had little effect on financial markets .]

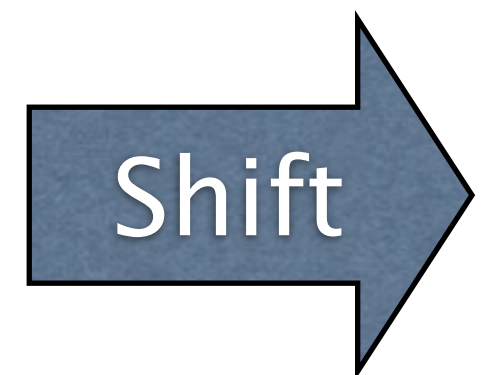
σ [root]

Example:

Arc-Eager

β [economic news had little effect on financial markets .]

σ [root]



Example:

Arc-Eager

β [news had little effect on financial markets .]

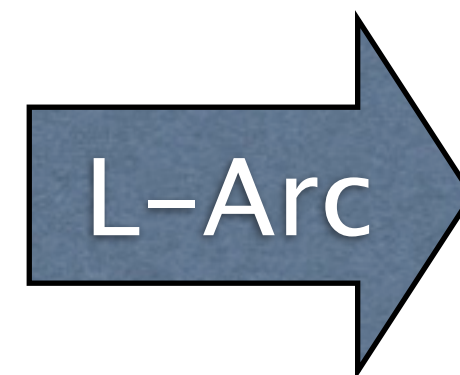
σ [root, economic]

Example:

Arc-Eager

β [news had little effect on financial markets .]

σ [root, economic]



Example:

Arc-Eager

β [news had little effect on financial markets .]

σ [root]

news
|
economic

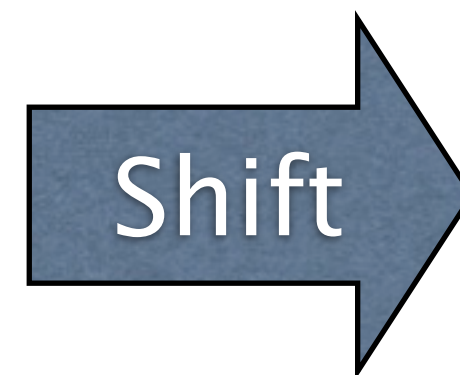
Example:

Arc-Eager

β [news had little effect on financial markets .]

σ [root]

news
|
economic



Example:

Arc-Eager

β [had little effect on financial markets .]

σ [root, news]

news
|
economic

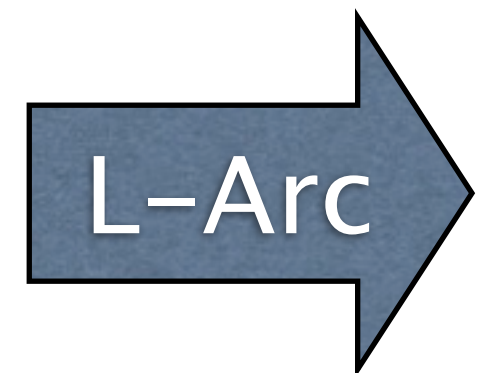
Example:

Arc-Eager

β [had little effect on financial markets .]

σ [root, news]

news
|
economic



Example:

Arc-Eager

β [had little effect on financial markets .]

σ [root]

had
|
news
|
economic

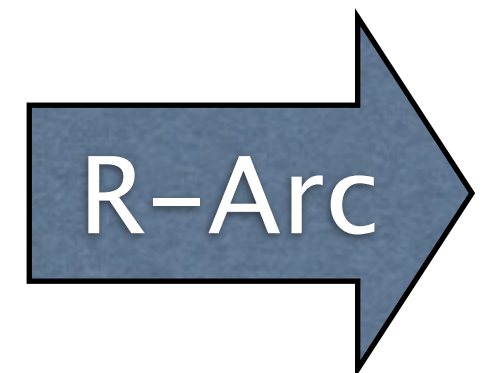
Example:

Arc-Eager

β [had little effect on financial markets .]

σ [root]

had
|
news
|
economic



Example:

Arc-Eager

β [little effect on financial markets .]

σ [root, had]

root
|
had
|
news
|
economic

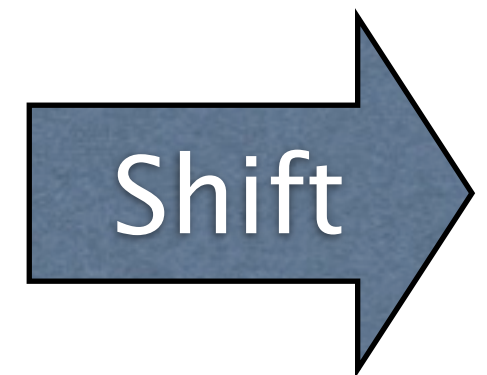
Example:

Arc-Eager

β [little effect on financial markets .]

σ [root, had]

root
|
had
|
news
|
economic



Example:

Arc-Eager

β [effect on financial markets .]

σ [root, had, little]

root
|
had
|
news
|
economic

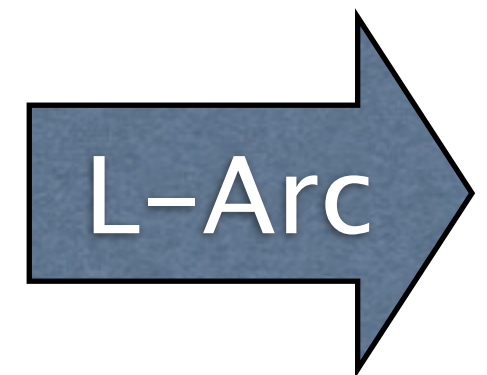
Example:

Arc-Eager

β [effect on financial markets .]

σ [root, had, little]

root
|
had
|
news
|
economic

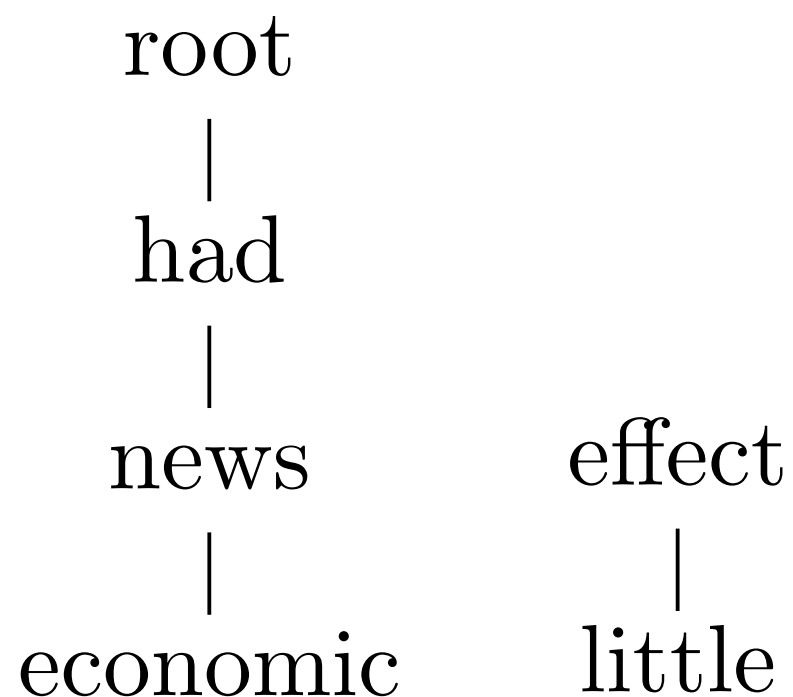


Example:

Arc-Eager

β [effect on financial markets .]

σ [root, had]



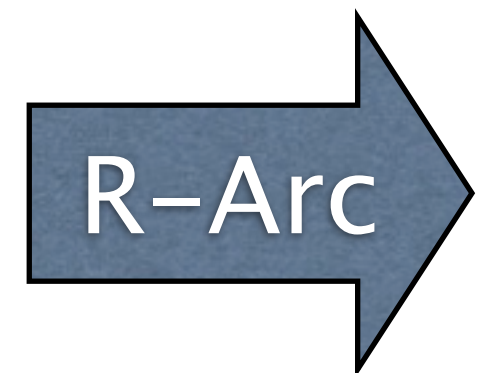
Example:

Arc-Eager

β [effect on financial markets .]

σ [root, had]

root	
had	
news	effect
economic	little

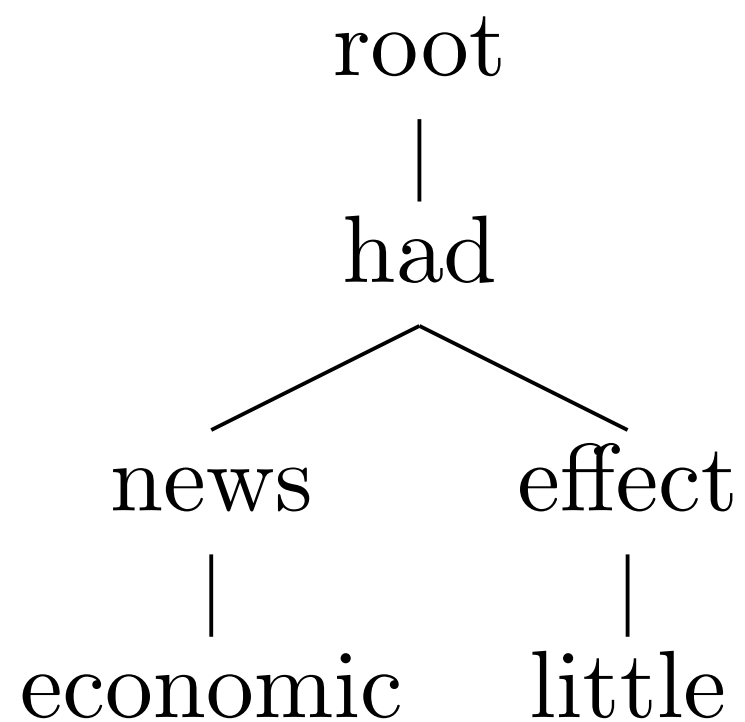


Example:

Arc-Eager

β [on financial markets .]

σ [root, had, effect]

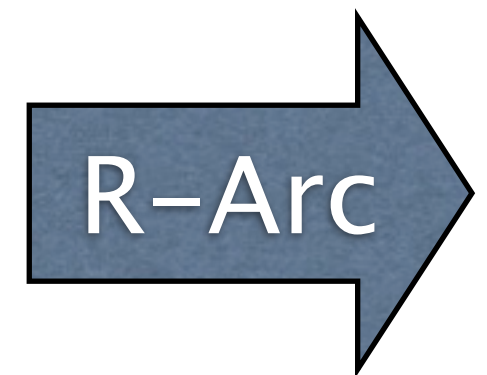
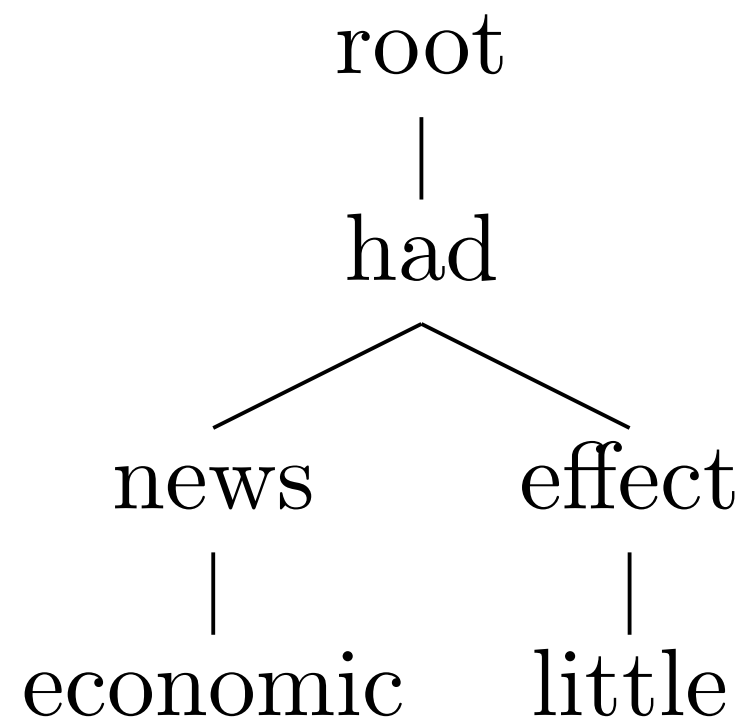


Example:

Arc-Eager

β [on financial markets .]

σ [root, had, effect]

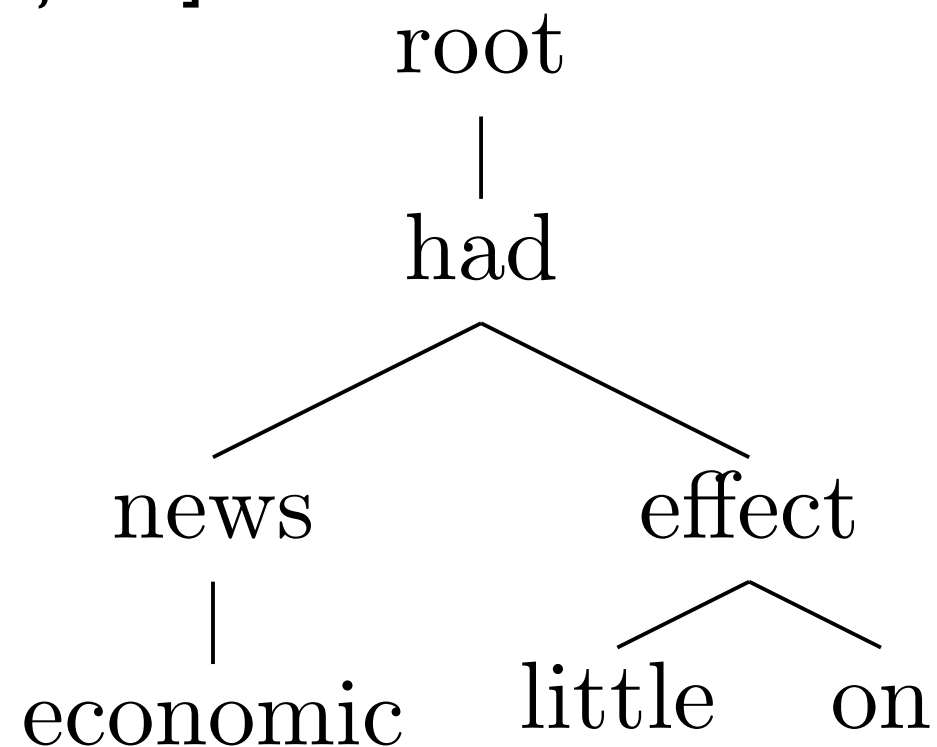


Example:

Arc-Eager

β [financial markets .]

σ [root, had, effect, on]

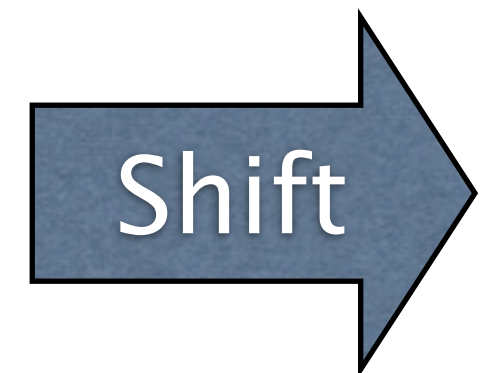
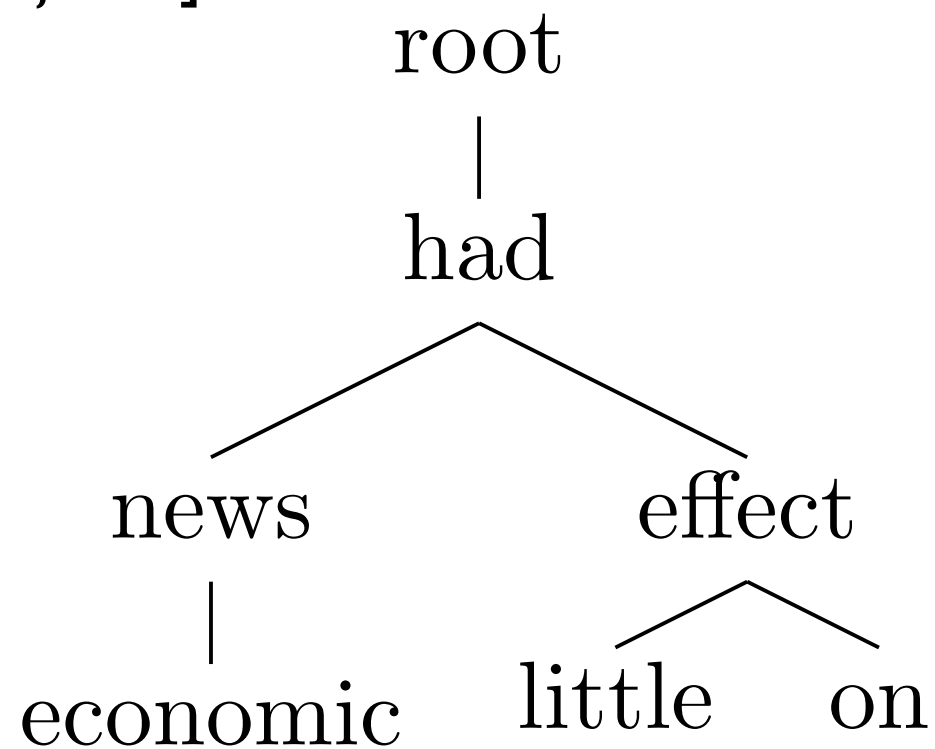


Example:

Arc-Eager

β [financial markets .]

σ [root, had, effect, on]

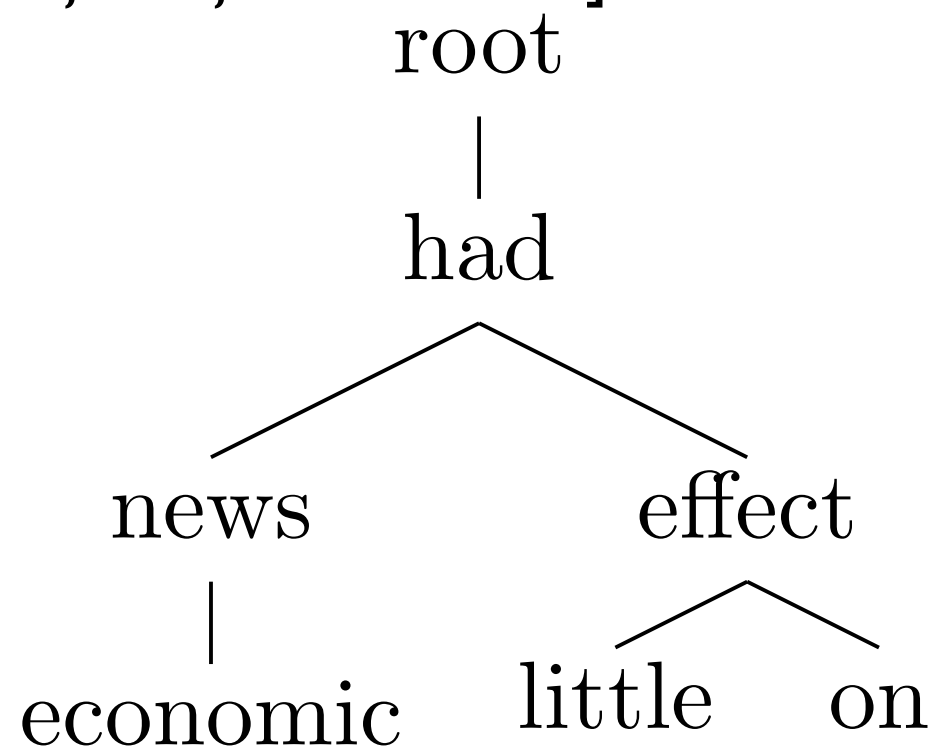


Example:

Arc-Eager

β [markets .]

σ [root, had, effect, on, financial]

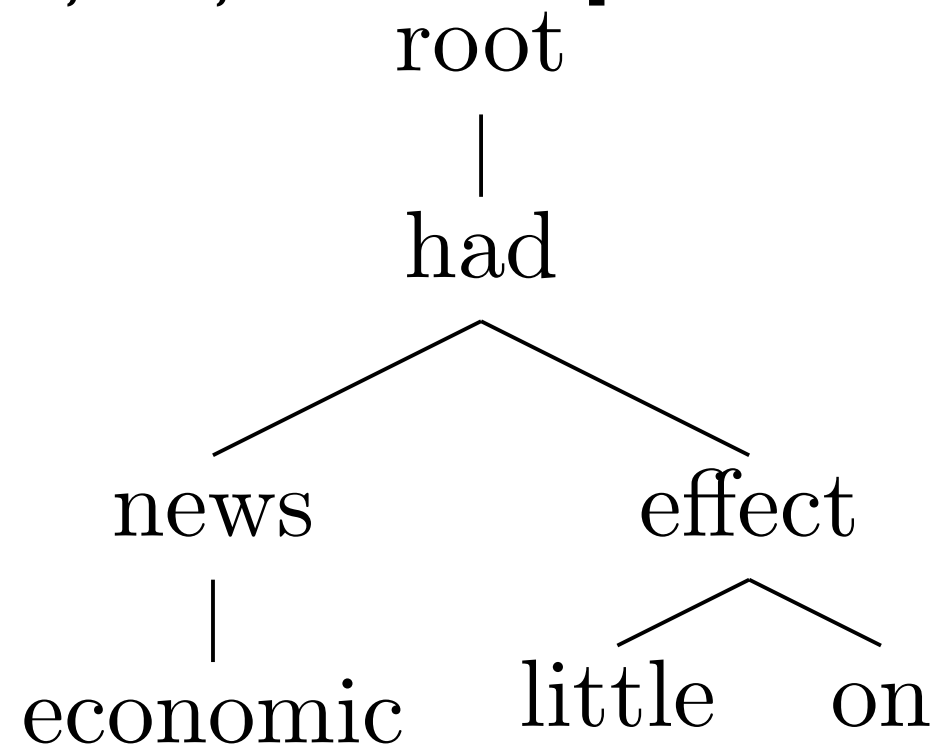


Example:

Arc-Eager

β [markets .]

σ [root, had, effect, on, financial]



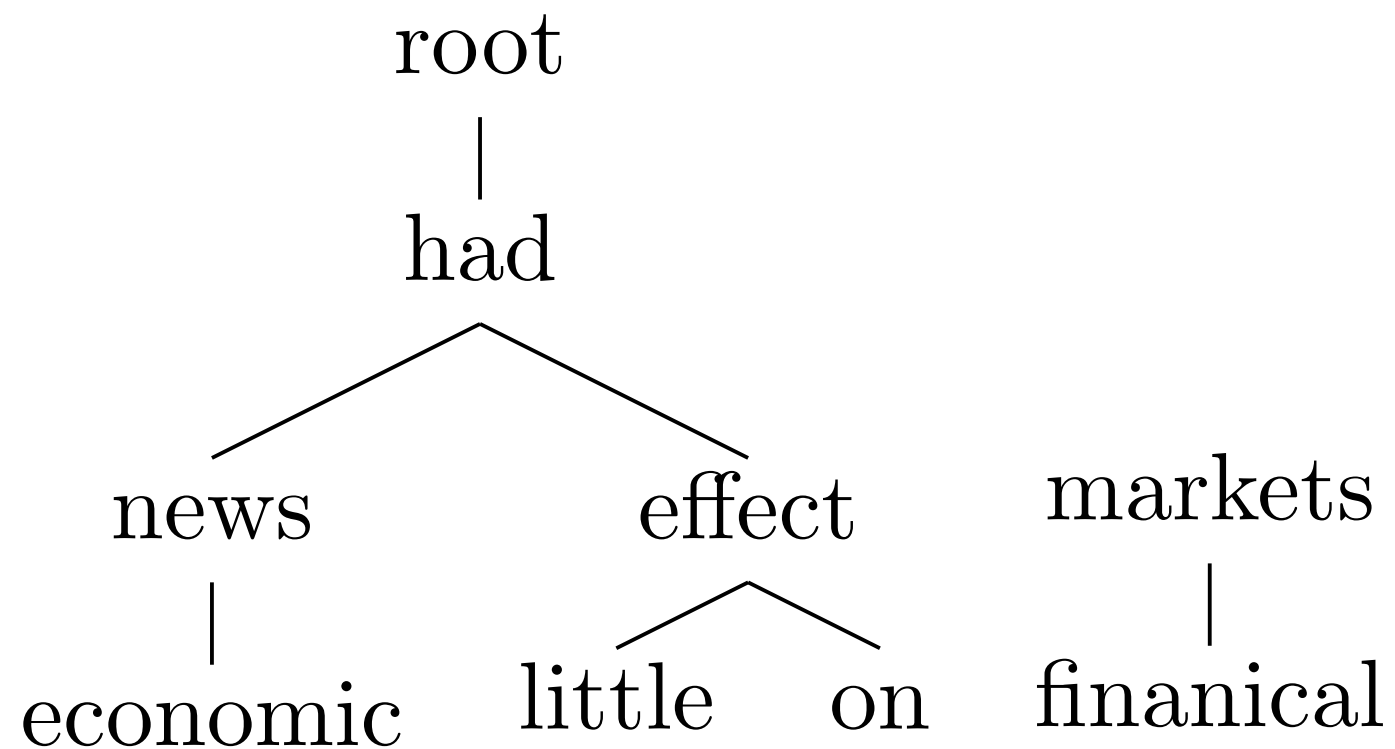
L-Arc

Example:

Arc-Eager

β [markets .]

σ [root, had, effect, on]

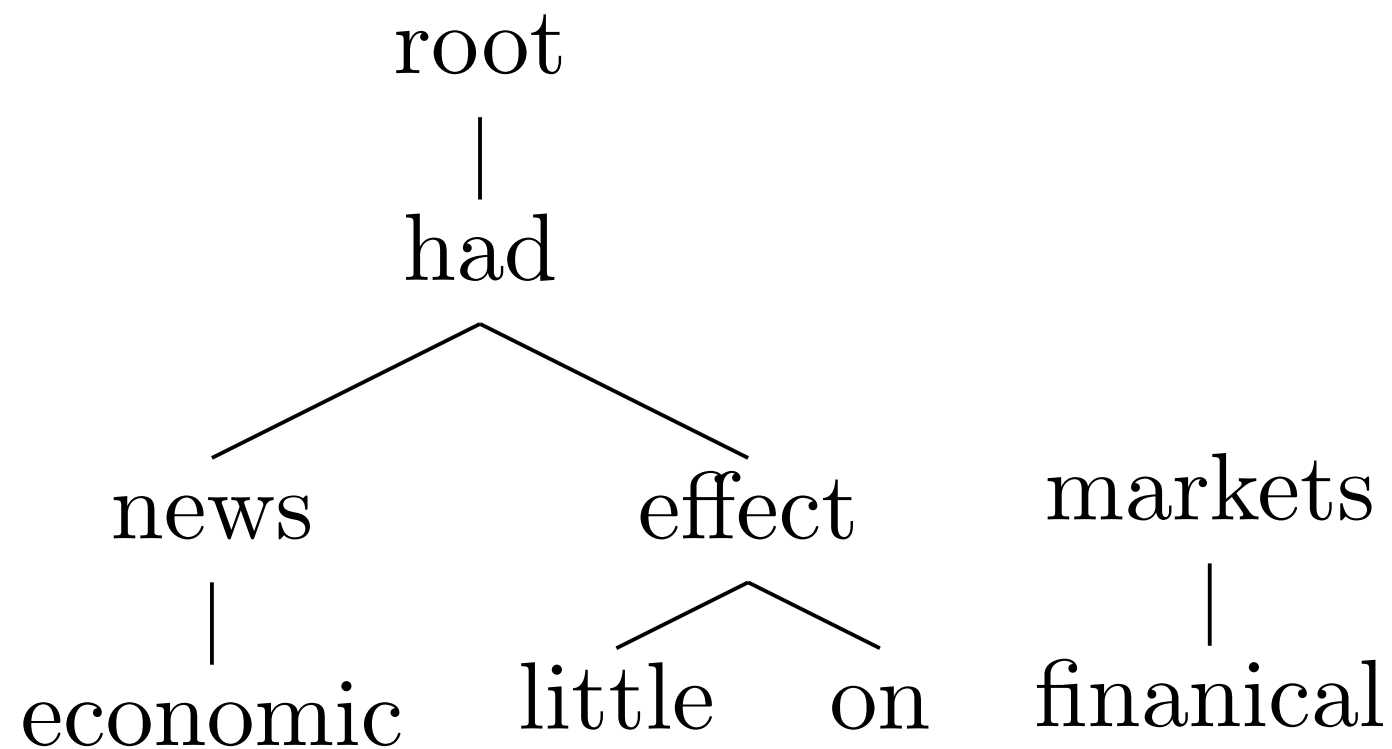


Example:

Arc-Eager

β [markets .]

σ [root, had, effect, on]



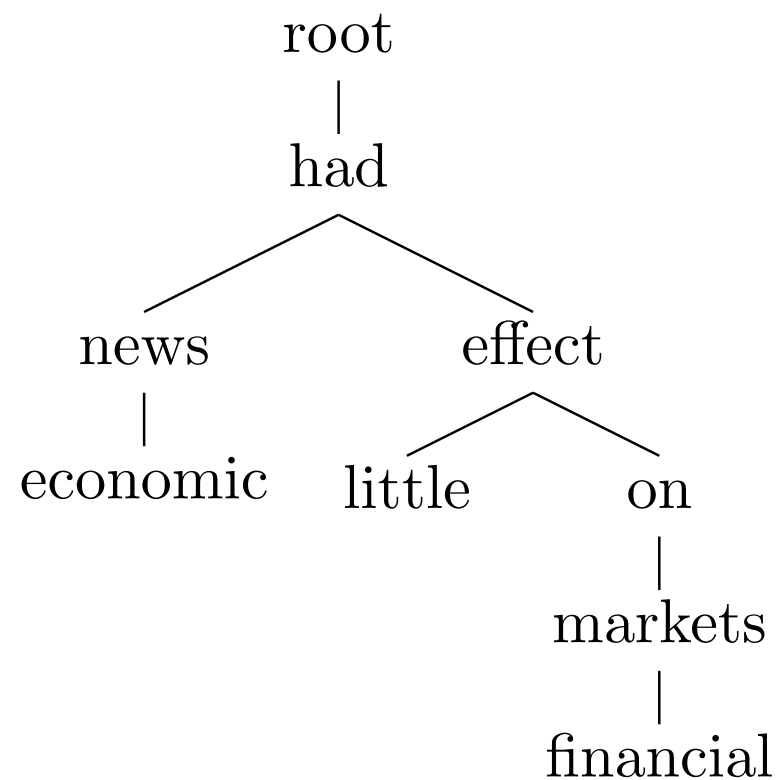
R-Arc

Example:

Arc-Eager

β [.]

σ [root, had, effect, on, markets]

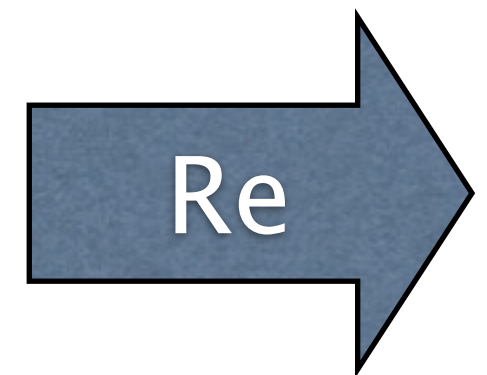
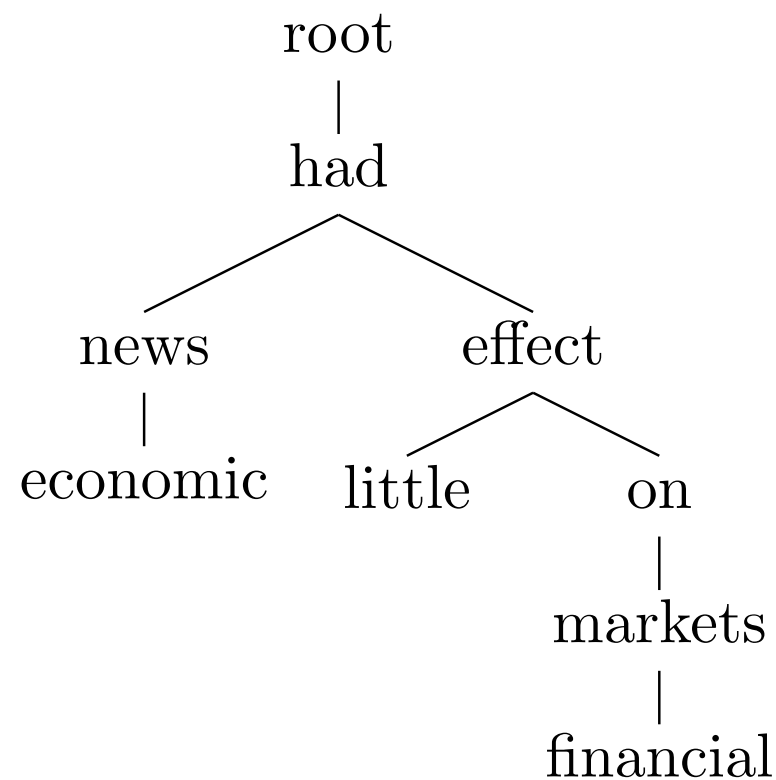


Example:

Arc-Eager

β [.]

σ [root, had, effect, on, markets]

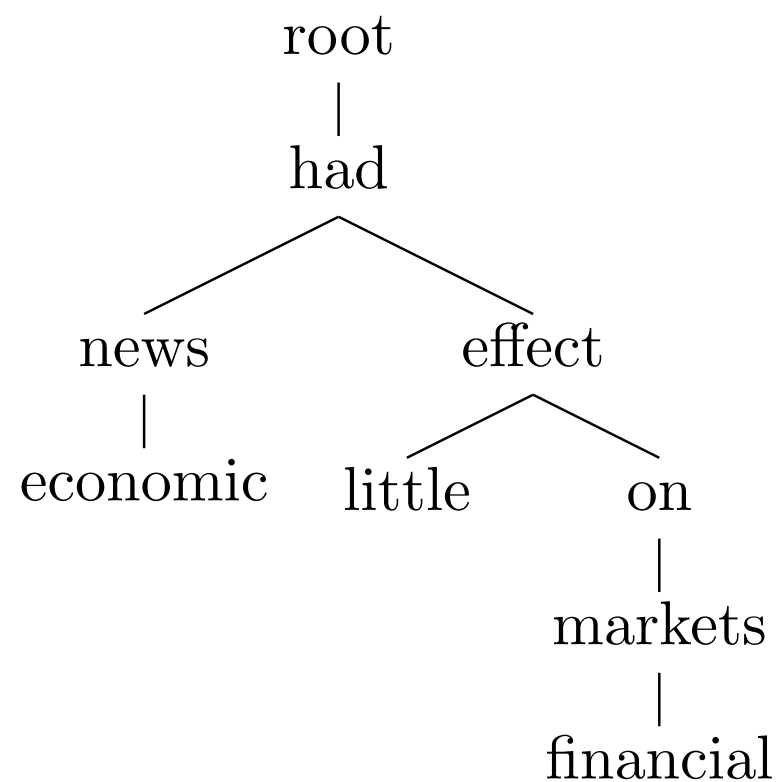


Example:

Arc-Eager

β [.]

σ [root, had, effect, on]

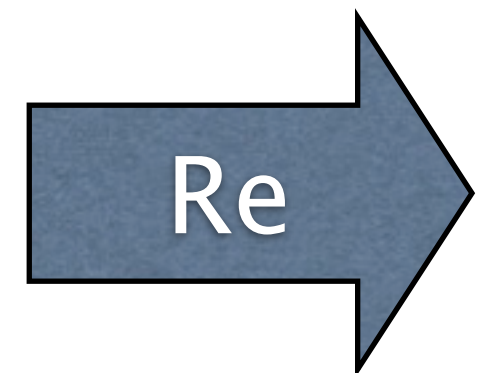
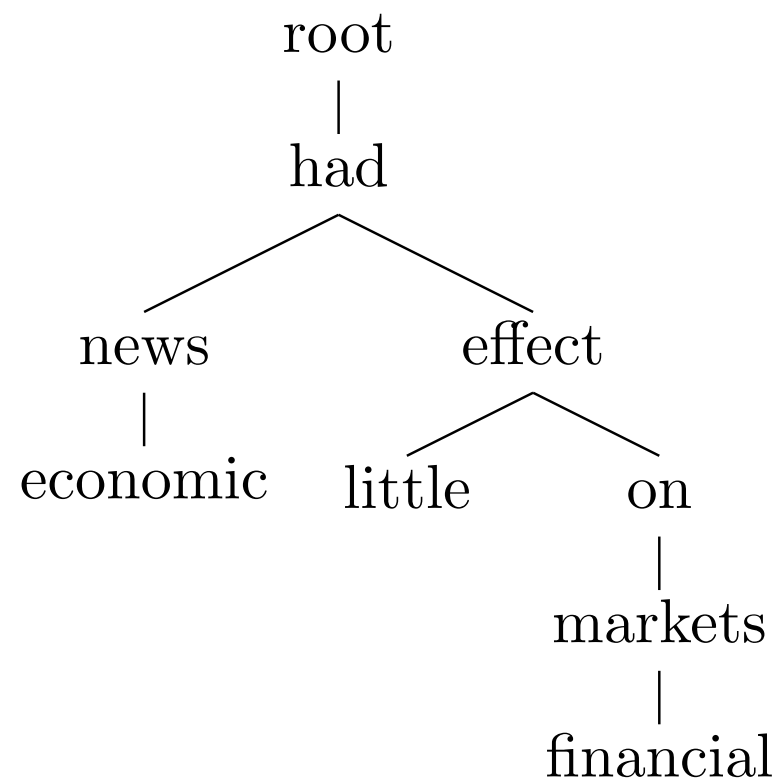


Example:

Arc-Eager

β [.]

σ [root, had, effect, on]

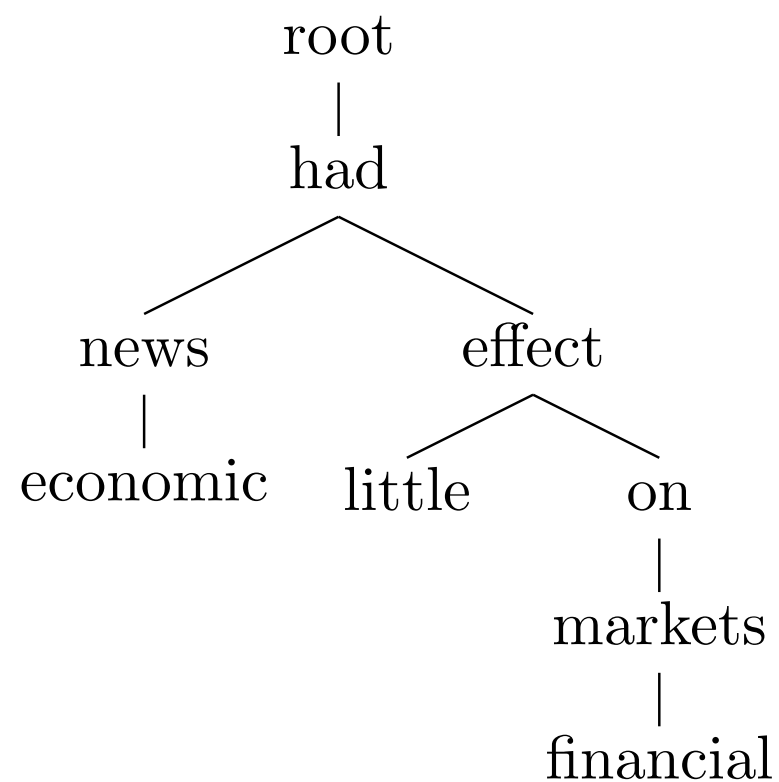


Example:

Arc-Eager

β [.]

σ [root, had, effect]

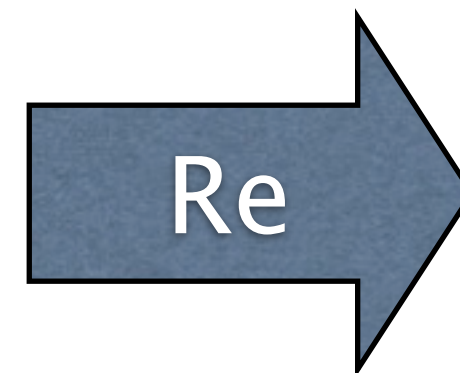
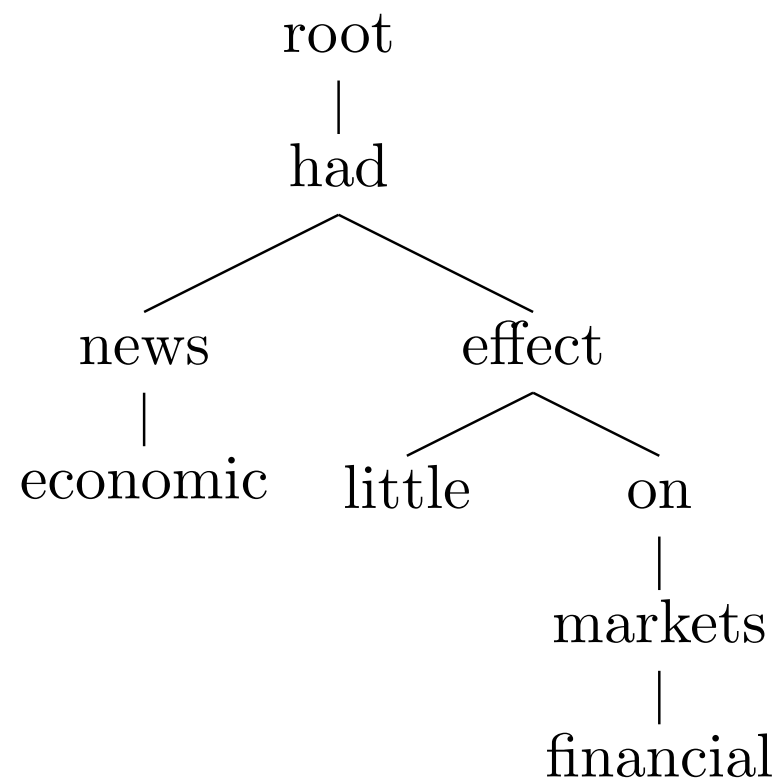


Example:

Arc-Eager

β [.]

σ [root, had, effect]

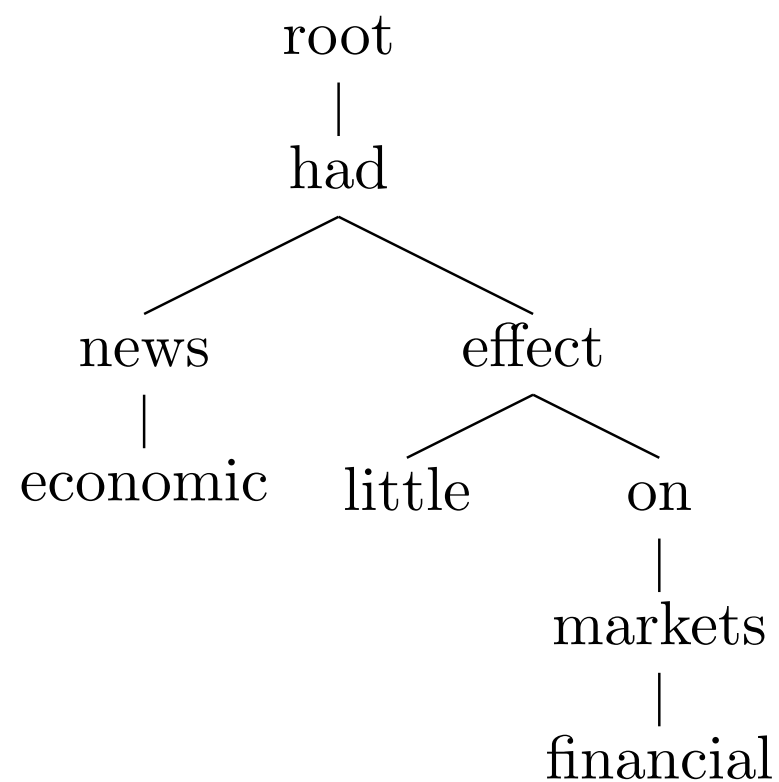


Example:

Arc-Eager

β [.]

σ [root, had]

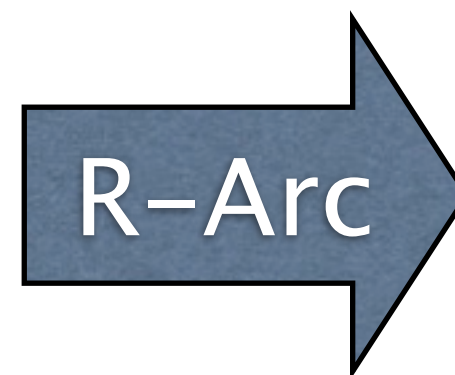
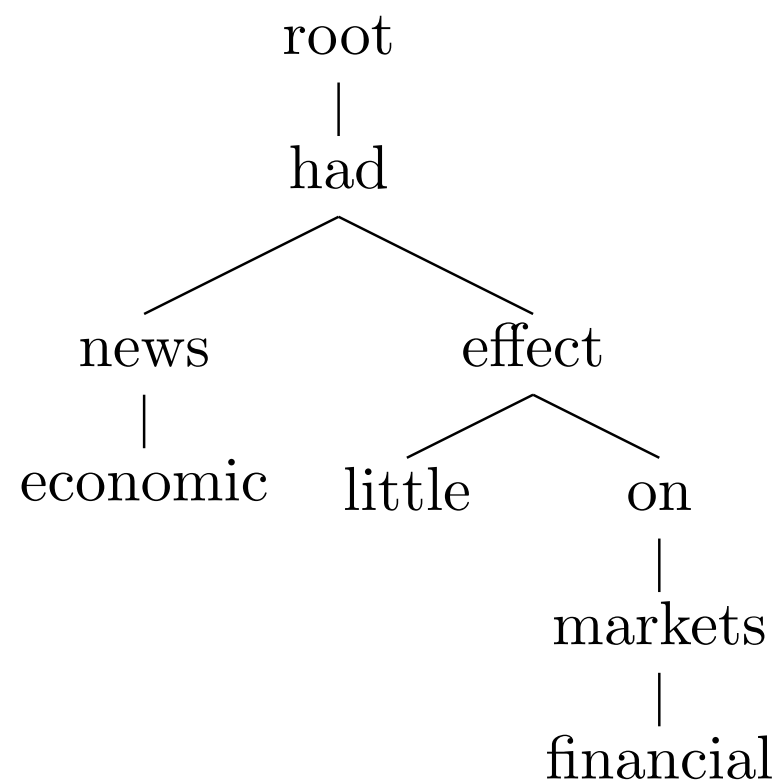


Example:

Arc-Eager

β [.]

σ [root, had]

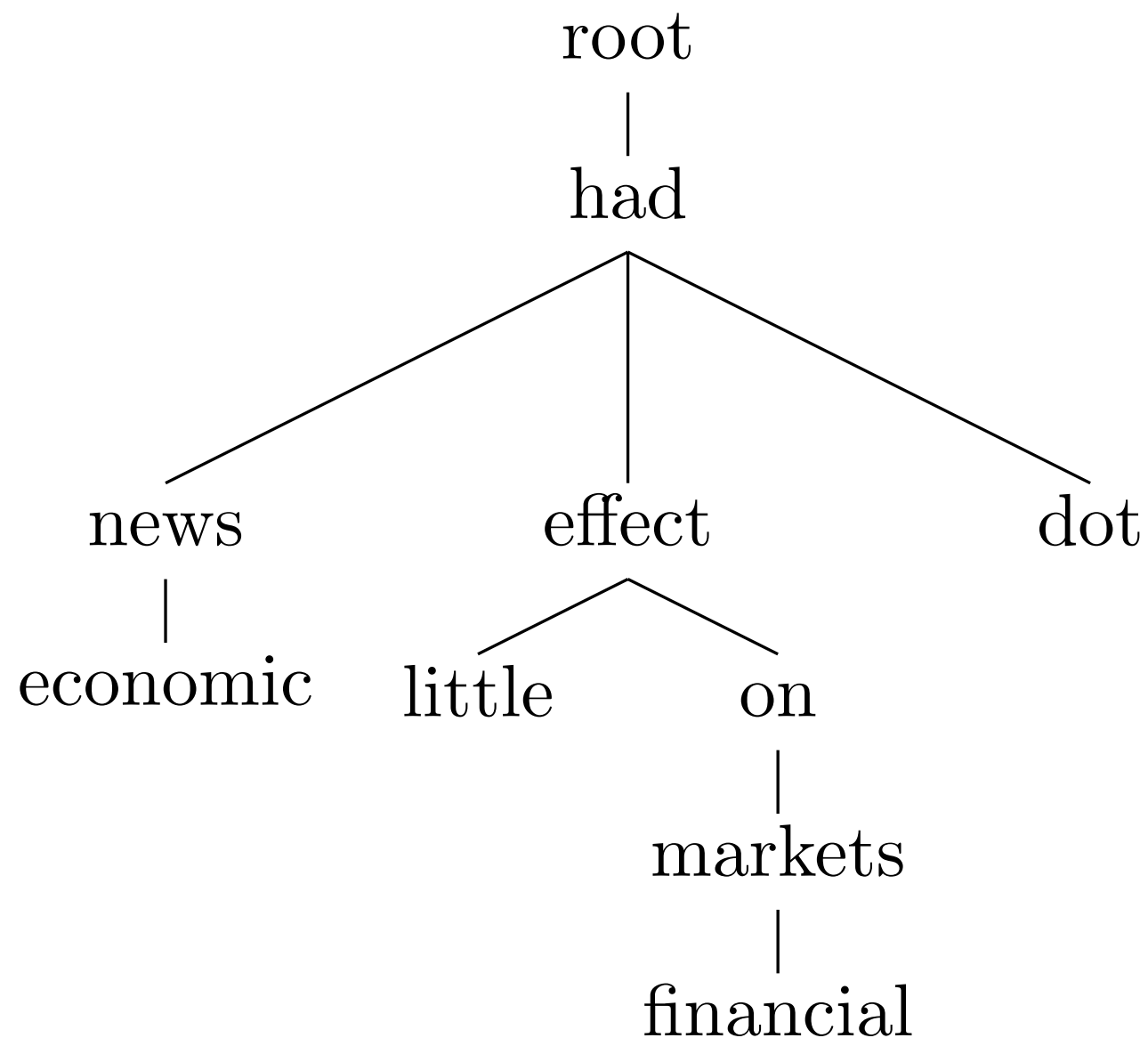


Example:

Arc-Eager

β []

σ [root, had, .]



Done

The Oracle

$$O : \mathcal{C} \rightarrow \mathcal{A}$$

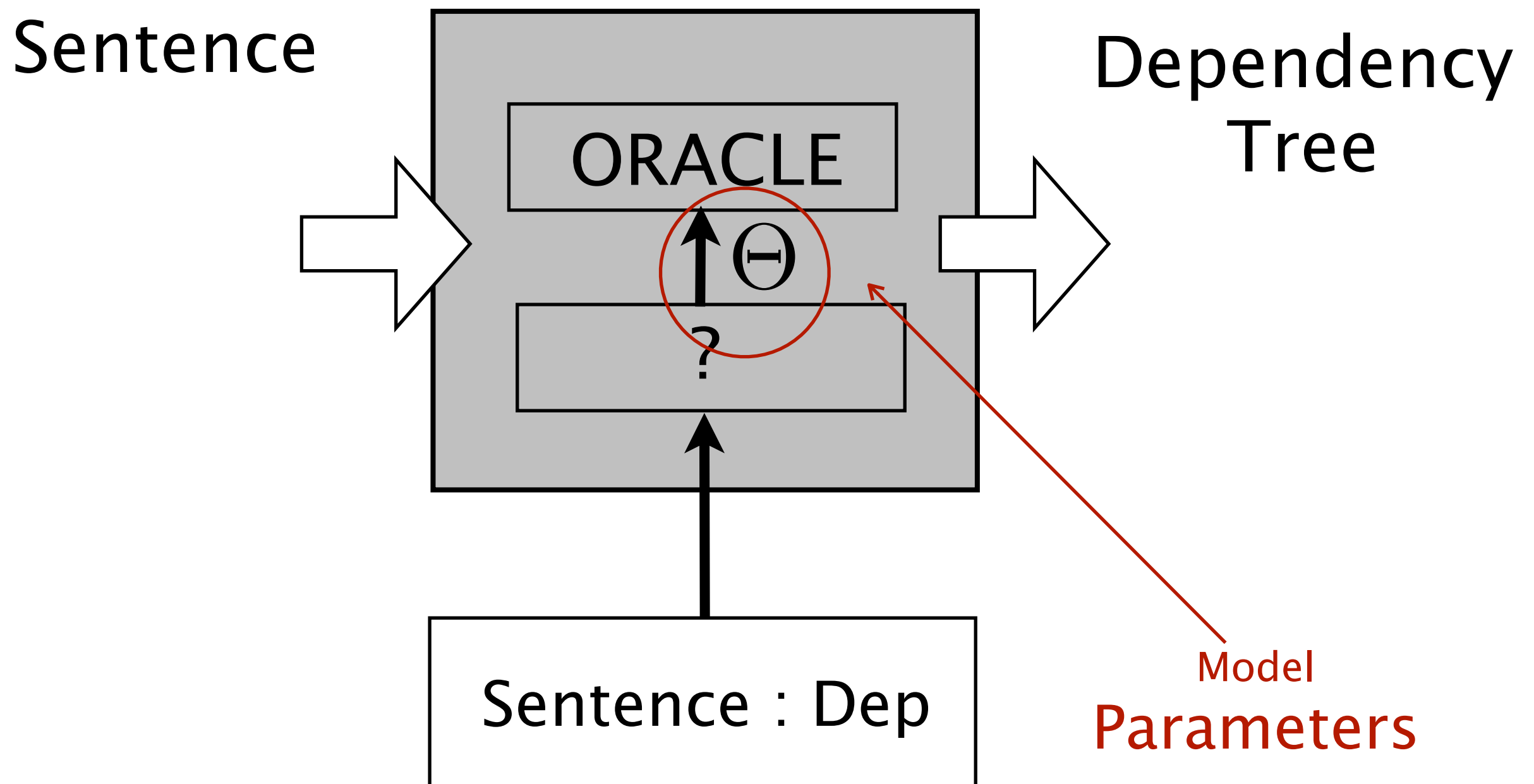
- Arc-Standard
- Arc-Eager
- Covington
- Nivre (and More, See Nivre 2008)

<http://www.aclweb.org/anthology/J/J08/J08-4003.pdf>

Transition-Based Parsing

- Representation: Dependency Trees
- Model: Transition-Based
- Inference: Oracle, Deterministic
- Learning: ?
- Evaluation: Labeled/Unlabeled AS

Transition-Based Parsing



Classifier-Based Learning

- Model Parameters, Take 1:

$$f : \mathcal{C} \times \mathcal{A} \rightarrow \mathcal{R}$$

- Model Parameters, Take 2:

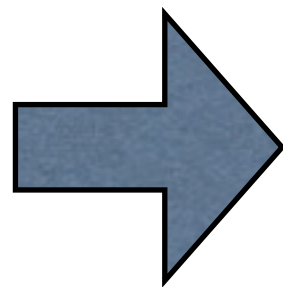
$$f : \phi(\mathcal{C}) \times \mathcal{A} \rightarrow \mathcal{R}$$

Feature Vector

[economic news had little effect on financial markets .]

[root]

[]



Buffer[0]	Form
Buffer[1]	Form
Stack[0]	Form
Stack[0]	POS
Stack[0]	Gender?
Stack[1]	Gender?
Stack[1]	POS
LDep[S[0]]	DepRel
LDep[S[0]]	Feminine?
RDep[S[0]]	Feminine?

$f(c_0) =$

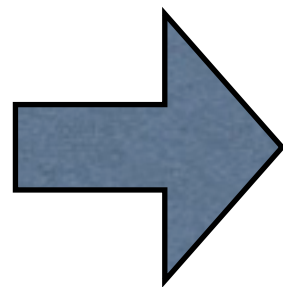
[economic, news, root, NULL, NULL, NULL, NULL, NULL, NULL, NULL]

Feature Vector

β [economic news had little effect on financial markets .]

[root]

[]



Buffer[0]	Form
Buffer[1]	Form
Stack[0]	Form
Stack[0]	POS
Stack[0]	Gender?
Stack[1]	Gender?
Stack[1]	POS
LDep[S[0]]	DepRel
LDep[S[0]]	Feminine?
RDep[S[0]]	Feminine?

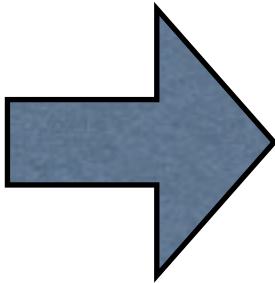
$f(c_0) =$

[economic, news, root, NULL, NULL, NULL, NULL, NULL, NULL, NULL]

Feature Vector

β [economic news had little effect on financial markets .]

σ [root]
[]



Buffer[0]	Form
Buffer[1]	Form
Stack[0]	Form
Stack[0]	POS
Stack[0]	Gender?
Stack[1]	Gender?
Stack[1]	POS
LDep[S[0]]	DepRel
LDep[S[0]]	Feminine?
RDep[S[0]]	Feminine?

$f(c_0) =$

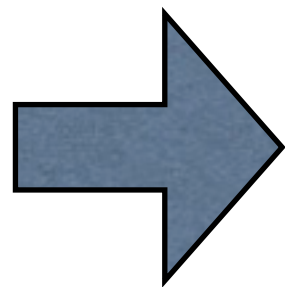
[economic, news, root, NULL, NULL, NULL, NULL, NULL, NULL, NULL]

Feature Vector

β [economic news had little effect on financial markets .]

σ [root]

A []

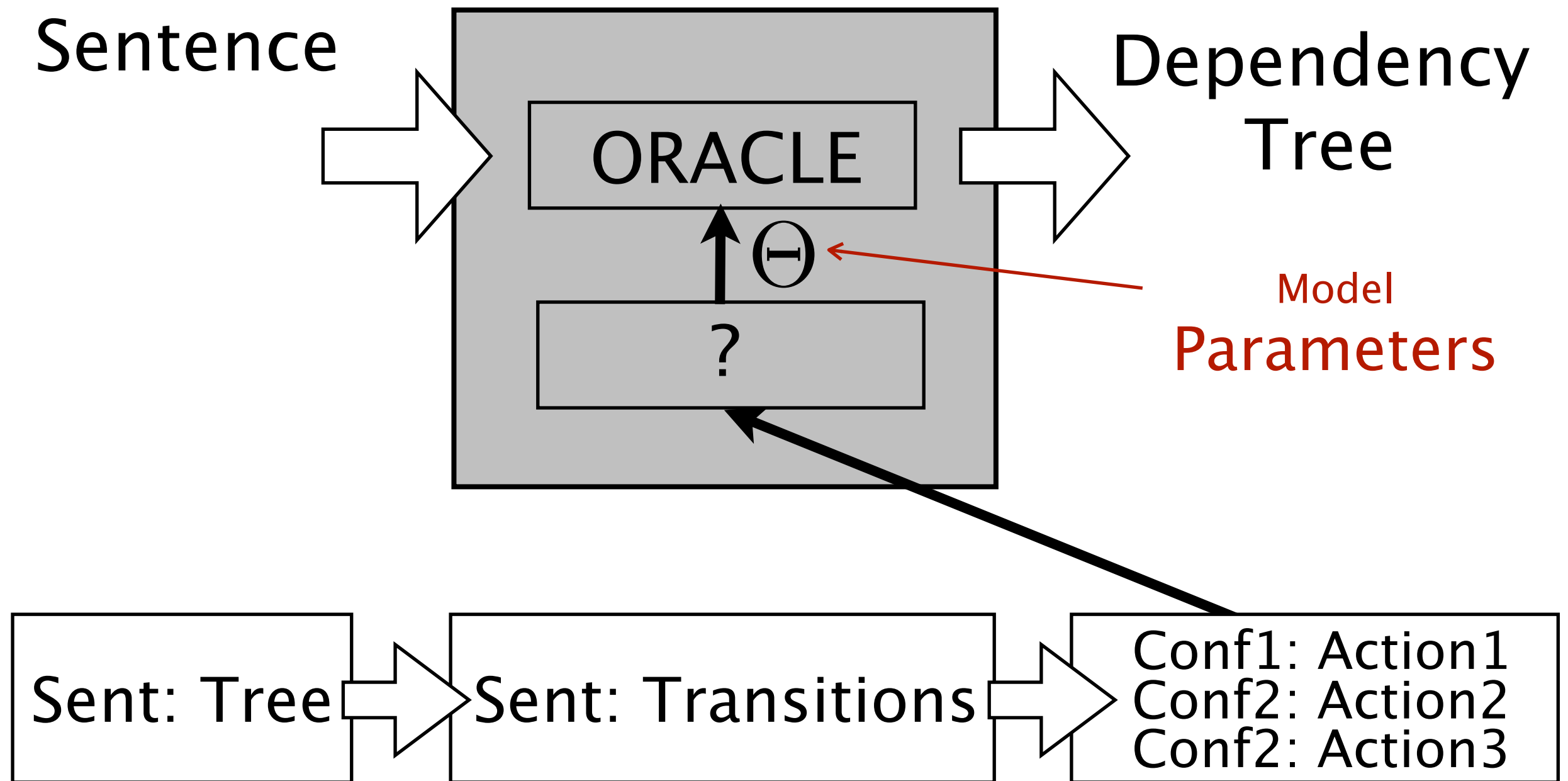


Buffer[0]	Form
Buffer[1]	Form
Stack[0]	Form
Stack[0]	POS
Stack[0]	Gender?
Stack[1]	Gender?
Stack[1]	POS
LDep[S[0]]	DepRel
LDep[S[0]]	Feminine?
RDep[S[0]]	Feminine?


$f(c_0) =$

[economic, news, root, NULL, NULL, NULL, NULL, NULL, NULL, NULL]

Transition-Based Parsing



Learning

$$a^* = \underset{\langle a' \rangle}{arg \max} \mathbf{w} \phi(c, a')$$


Learning a standard Linear Classifier

Memory-Based Learning
Support Vector Machines
The Perceptron Algorithm

Learning

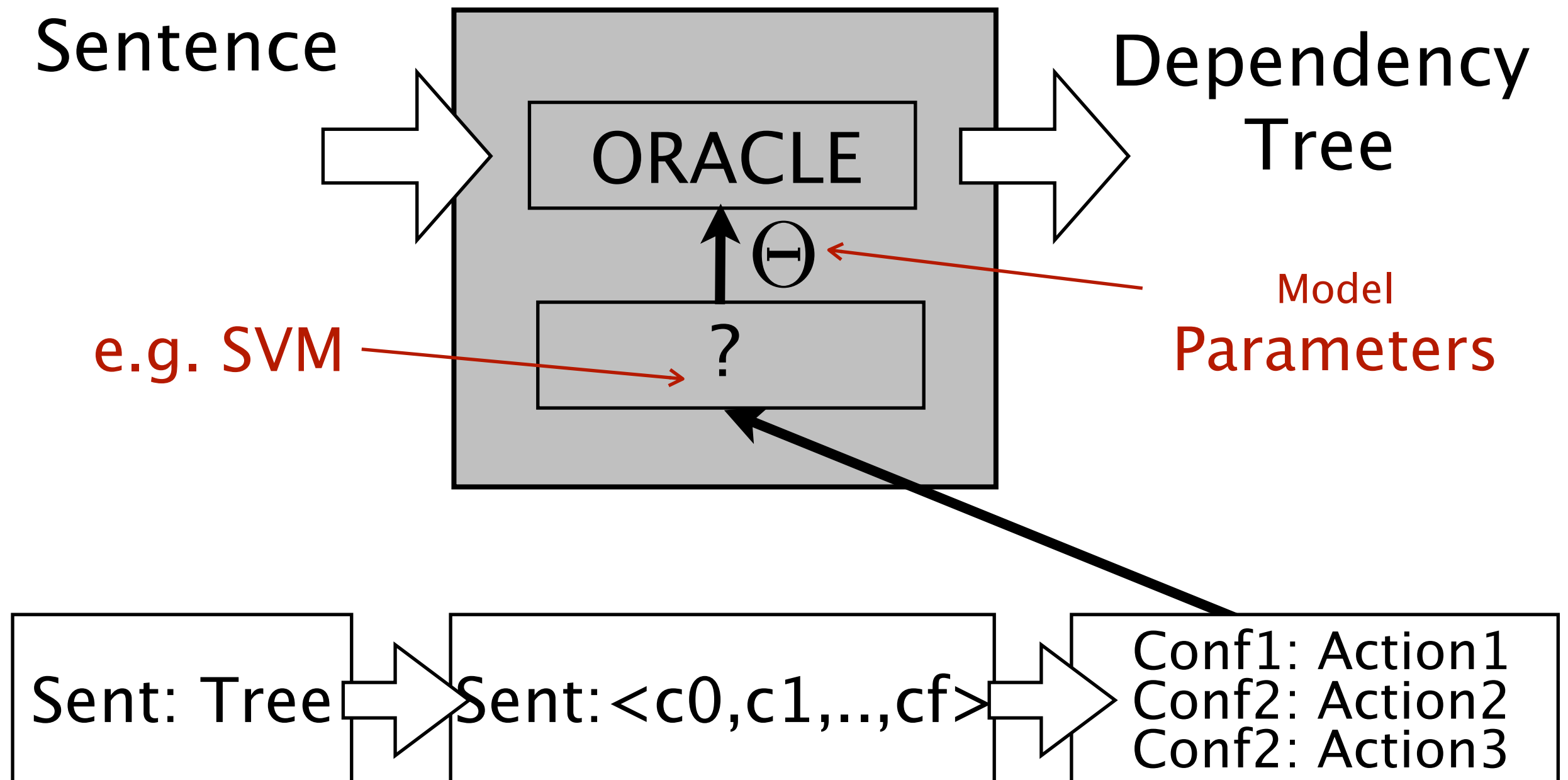
$$a^* = \underset{\langle a' \rangle}{arg \max} \mathbf{w} \phi(c, a')$$

Learning a standard Linear Classifier

Memory-Based Learning
Support Vector Machines
The Perceptron Algorithm

MaltParser

(Nivre 2007)



Transition-Based Parsing Impl.(1)

- Representation: Dependency Trees
- Model: Transition-Based
- Inference: Oracle, Deterministic
- Learning: Support Vector Machines
- Evaluation: Labeled/Unlabeled AS

Transition-Based Parsing Impl.(1)

- Representation: Dependency Trees
- Model: Transition-Based
- Inference: Oracle, Deterministic
- Learning: Support Vector Machines
- Evaluation: Labeled/Unlabeled AS

<http://www.maltparser.org>

Transition-Based Parsing Impl.(2)

- Representation: Dependency Trees
- Model: Transition-Based
- Inference: Beam-Search
- Learning: Structured Perceptron
- Evaluation: Labeled/Unlabeled AS

Transition-Based Parsing Impl.(2)

- Representation: Dependency Trees
- Model: Transition-Based
- Inference: Beam-Search
- Learning: Structured Perceptron
- Evaluation: Labeled/Unlabeled AS

Not Today. Think of it as K-Best Parsing

Transition-Based Parsing Impl.(2)

- Representation: Dependency Trees
- Model: Transition-Based
- Inference: Beam-Search
- Learning: Structured Perceptron
- Evaluation: Labeled/Unlabeled AS

Not Today. Think of it as K-Best Parsing

<http://www.sutd.edu.sg/cmsresource/faculty/yuezhang/cl11.pdf>

Introducing Transition Systems for **MRLS**

Models for DS: Transition Systems

- A Transition system contains the following components

- A Buffer

- A Stack

$$m \in \mathcal{S}$$

- A Set of Arcs

$$\subseteq \mathcal{S} \times R \times \mathcal{S}$$

A Partial analysis of an input sentence

Models for DS: Transition Systems

- A Transition system contains the following components

- A Buffer

 β

- A Stack

 $m \in \mathcal{S}$

- A Set of Arcs

 $\subseteq \mathcal{S} \times R \times \mathcal{S}$

A Partial analysis of an input sentence

Models for DS: Transition Systems

- A Transition system contains the following components

- A Buffer

 β

- A Stack

 σ $m \in \mathcal{S}$

- A Set of Arcs

 $\subseteq \mathcal{S} \times R \times \mathcal{S}$

A Partial analysis of an input sentence

Models for DS: Transition Systems

- A Transition system contains the following components

- A Buffer

 β

- A Stack

 σ $m \in \mathcal{S}$

- A Set of Arcs

 A $\subseteq \mathcal{S} \times R \times \mathcal{S}$

A Partial analysis of an input sentence

Models for DS: Transition Systems

- A Transition system contains the following components

● A Buffer	β	$w \in V$
● A Stack	σ	$m \in \mathcal{S}$
● A Set of Arcs	A	$\subseteq \mathcal{S} \times R \times \mathcal{S}$

A Partial analysis of an input sentence

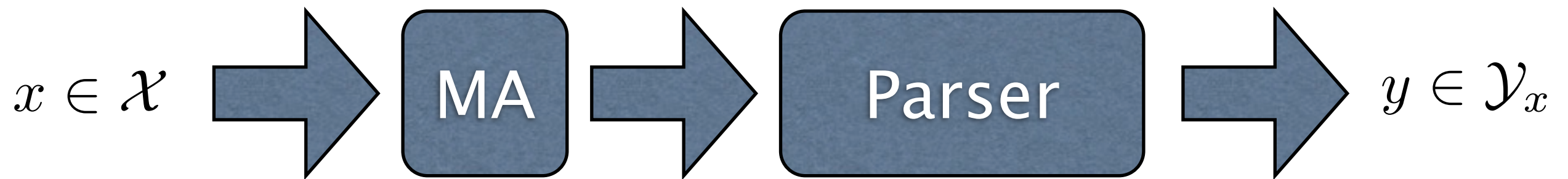
Two Possible Architectures

- Pipeline Architecture

- Joint Architecture

Two Possible Architectures

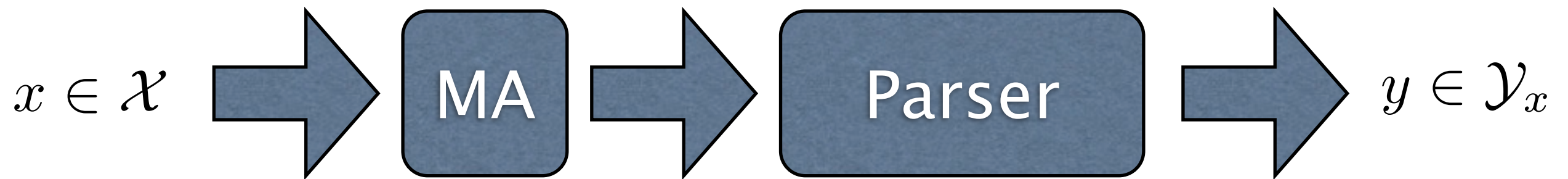
- Pipeline Architecture



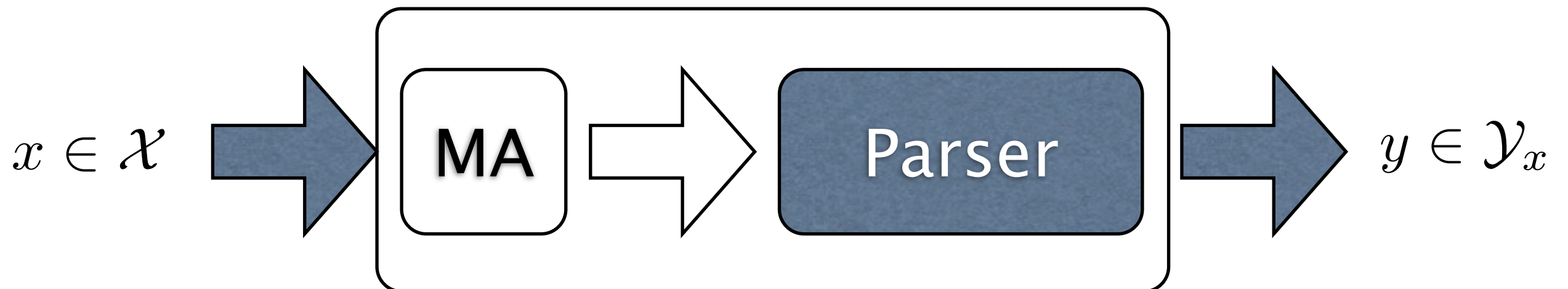
- Joint Architecture

Two Possible Architectures

- Pipeline Architecture



- Joint Architecture

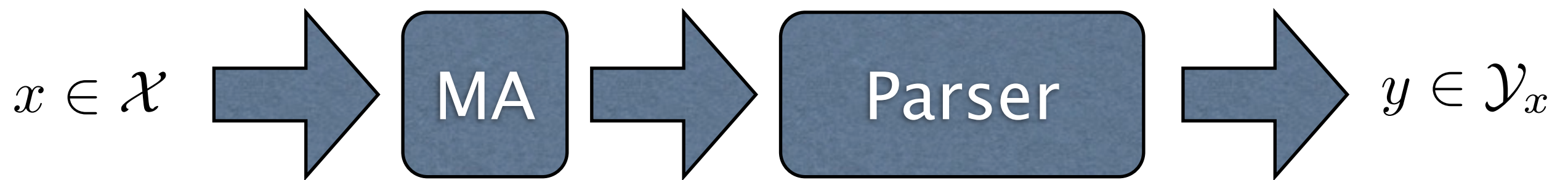


Inference

- So far: Pipeline Architectures only
- Two possible Scenarios
 - Gold MSR (optimistic)
 - Predicted MSR (realistic)

Inference

- So far: Pipeline Architectures only



- Two possible Scenarios
 - Gold MSR (optimistic)
 - Predicted MSR (realistic)

Learning

- Specially tailored feature vector
 - Look at morphological marking
 - Look at dependency labels
 - Correlate the morph-label
- Learn Different kinds of classifiers

Learning

- Specially tailored feature vector
 - Look at morphological marking
 - Look at dependency labels
 - Correlate the morph-label
- Learn Different kinds of classifiers

<http://nil.fdi.ucm.es/maltoptimizer/install.html>

Empirical Observations

- Arabic parsing benefitted from features like Case, Gender, Mood
- French benefitted from POS, Lemma
- Hindi benefitted from chunk-in cues
- Korean benefitted from "null" cues
- Noisy morphological information is worse than no morphology at all.

<http://www.tsarfaty.com/pdfs/spmrl10.pdf>

Shared Task 2007



Shared Task 2007

Low (76.31–76.94):

- Arabic, Basque, Greek

- Medium (79.19–80.21):

- Czech, Hungarian, Turkish

- High (84.40–89.61):

- Catalan, Chinese, English,



Non-Determinism

- Representation: Dependency Trees
- Model: Transition-Based
- Inference: Easy-First, dynamic
- Learning: Online, Perceptron
- Evaluation: Labeled/Unlabeled AS

Easy First Parsing

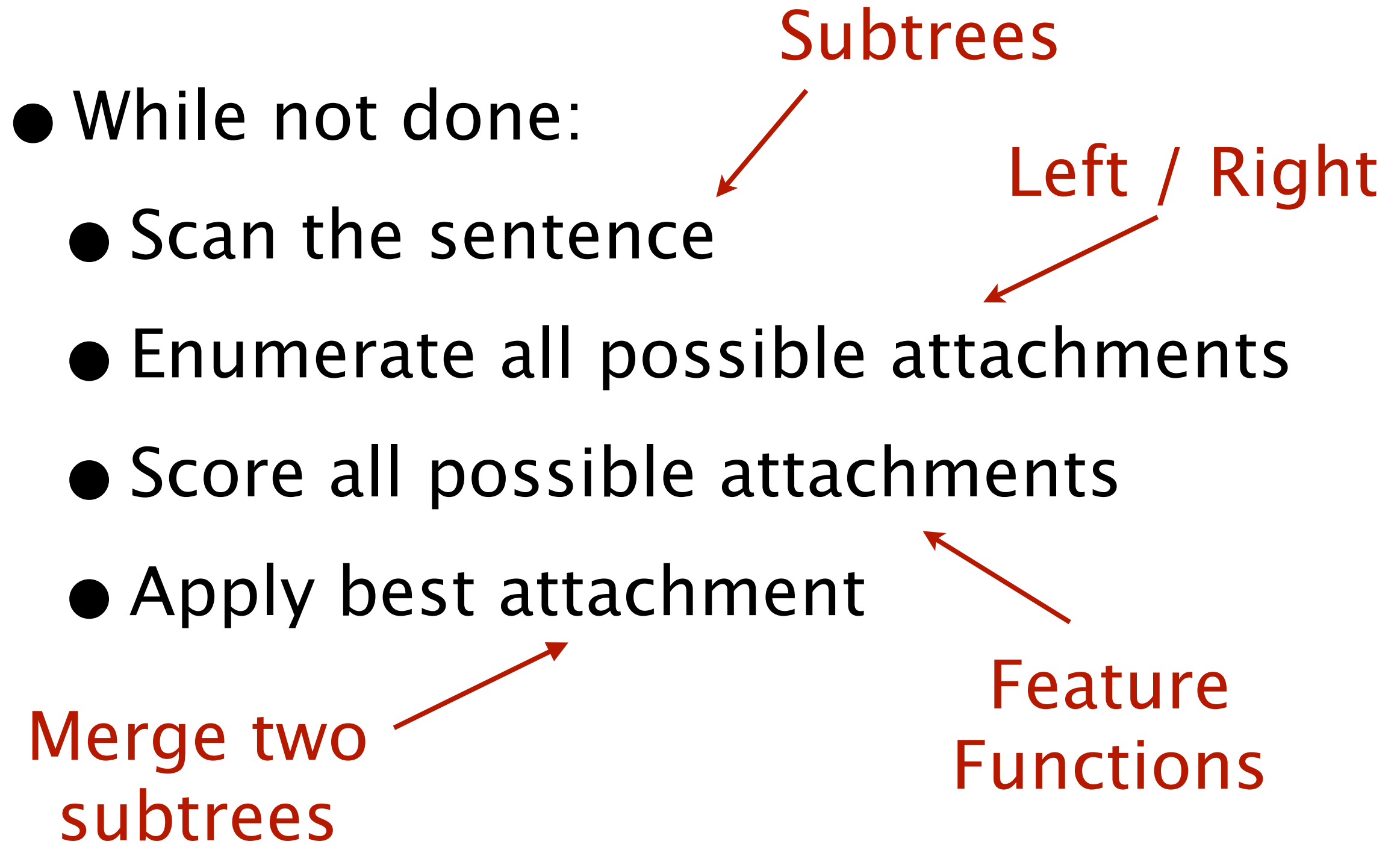
Goldberg and Elhadad 2010, 2011

Easy First Parsing

Goldberg and Elhadad 2010, 2011

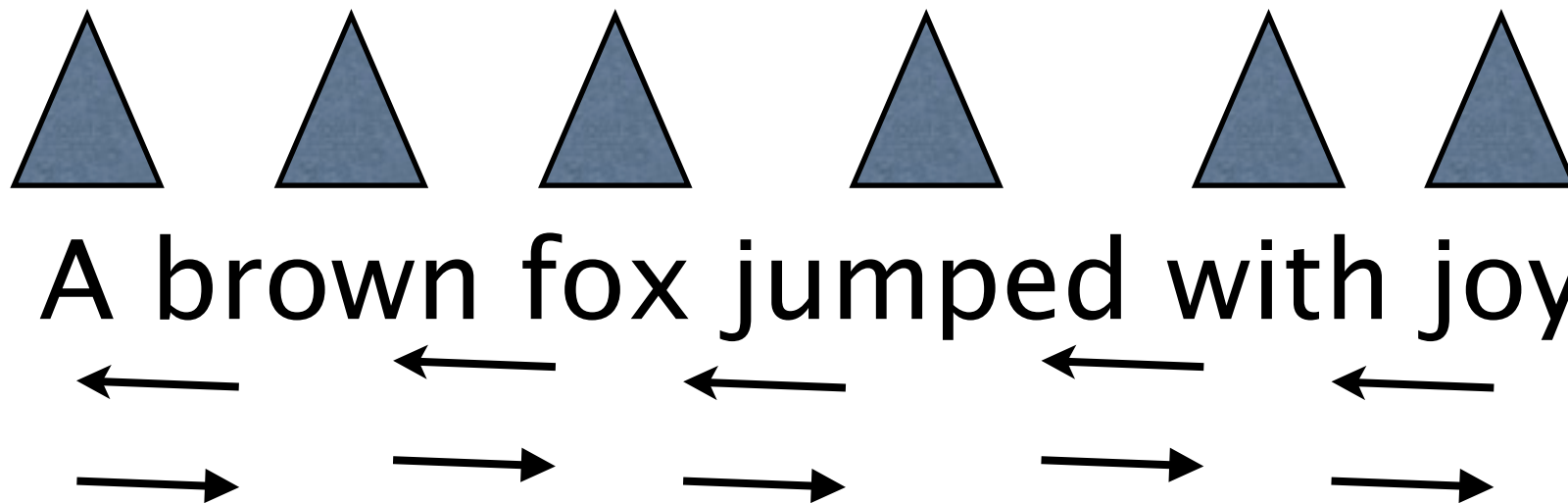
- Parsing order is dynamic
- Score according to context
- Make easy attachments first

Inference



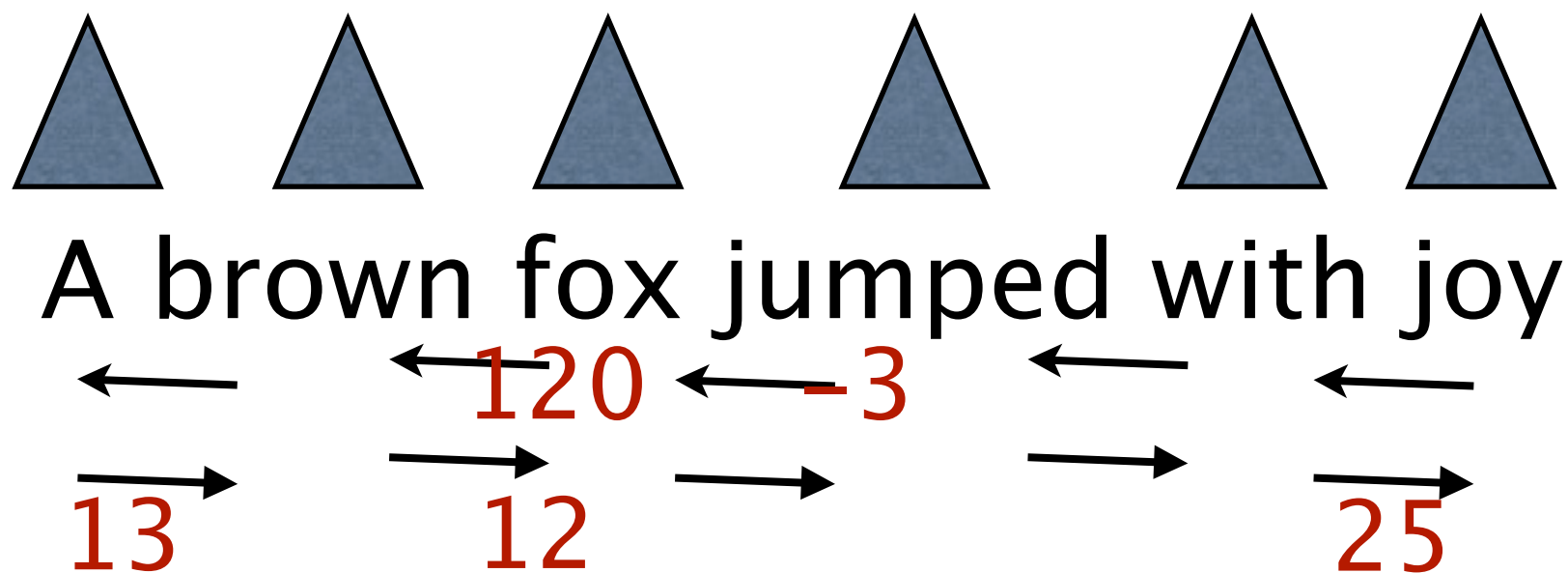
Example

Easy-First



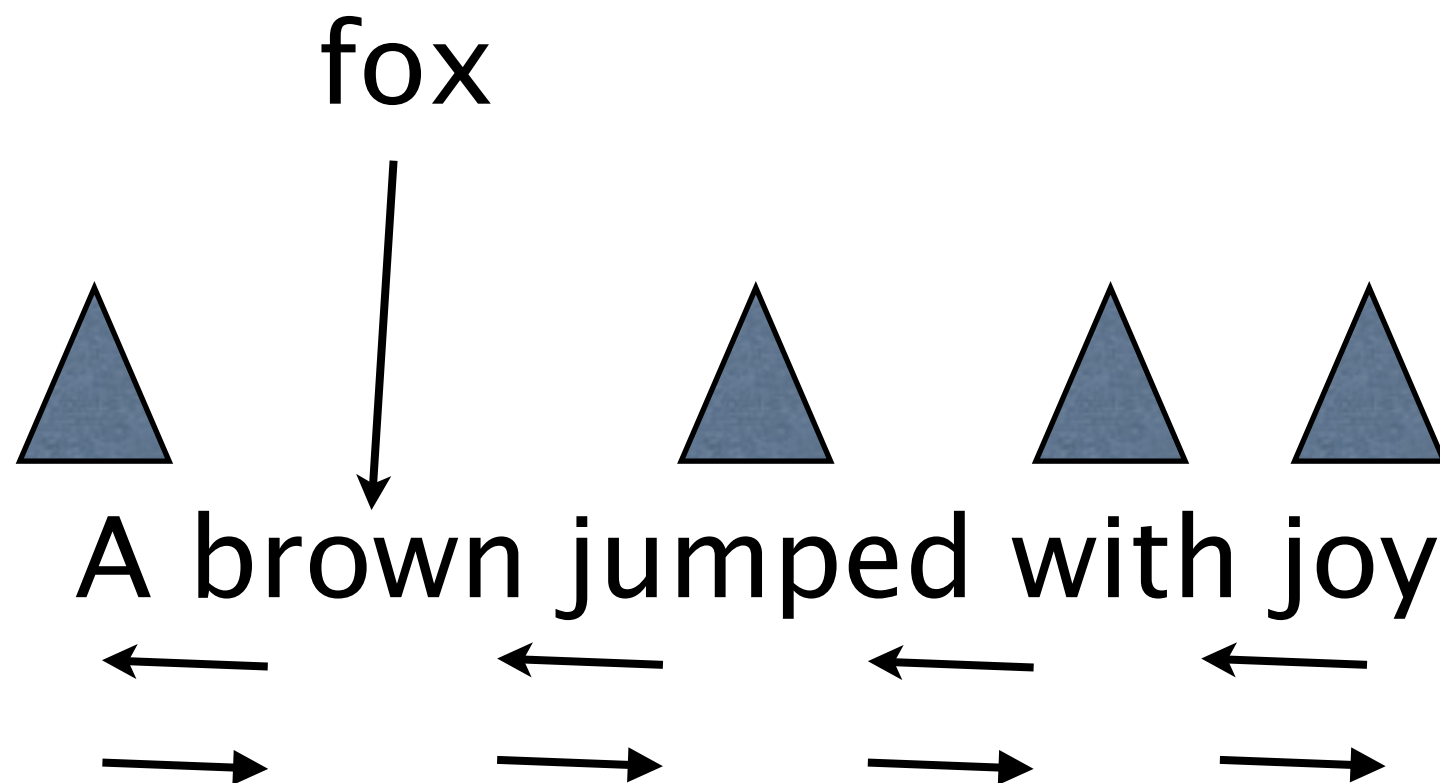
Example

Easy-First



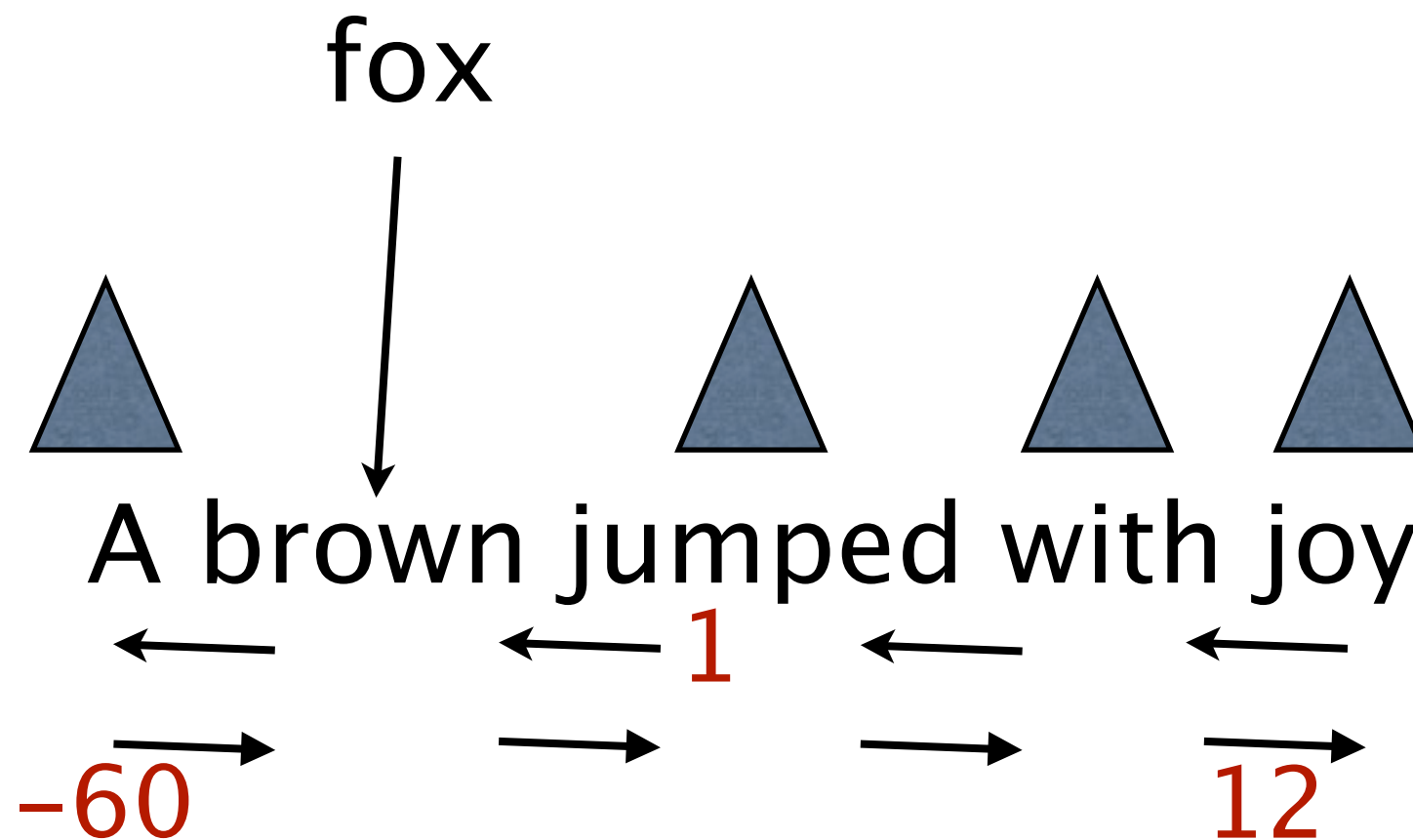
Example

Easy-First



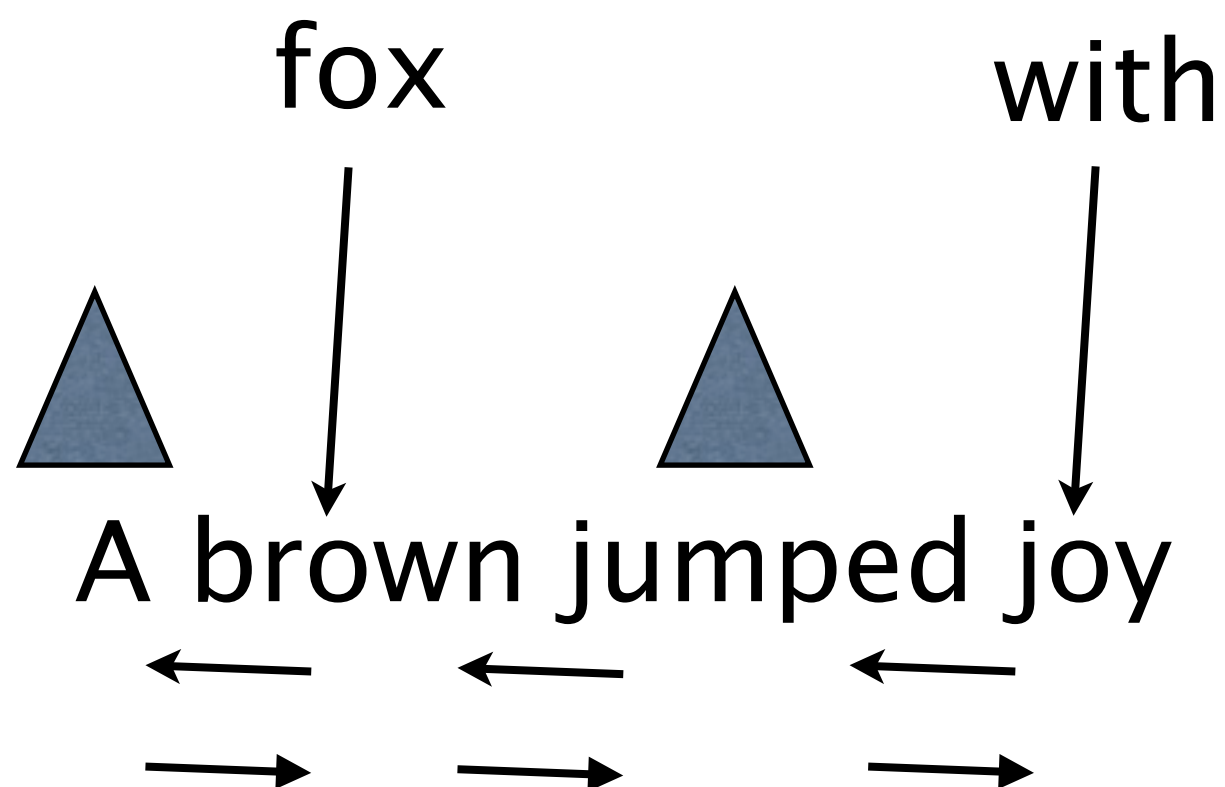
Example

Easy-First



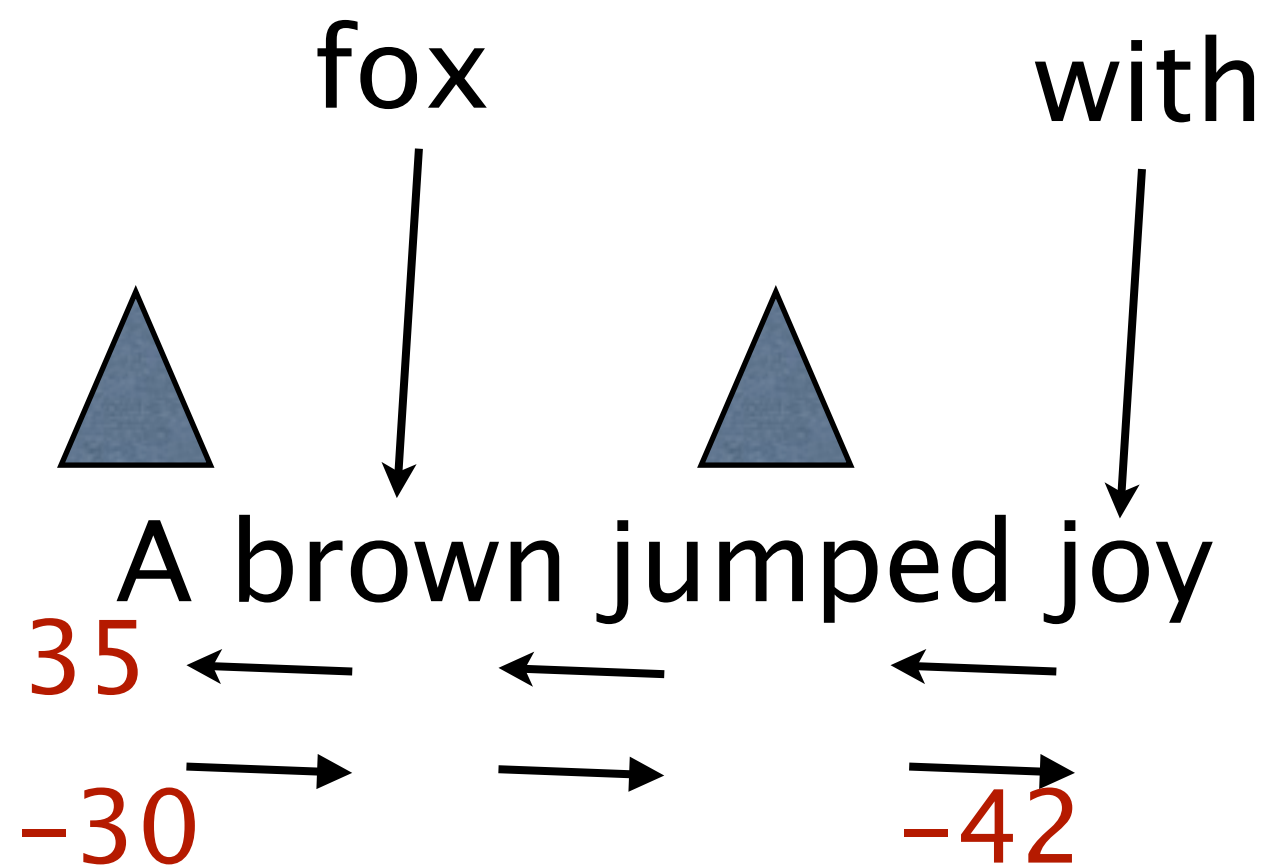
Example

Easy-First



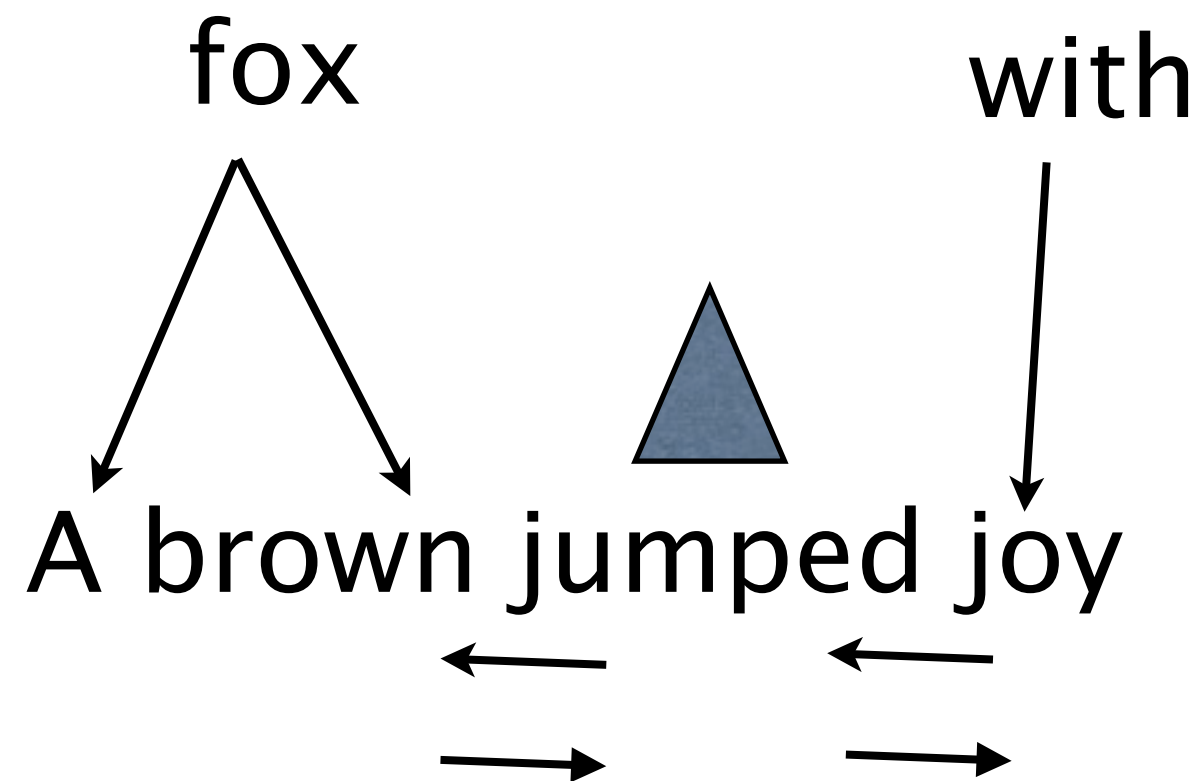
Example

Easy-First



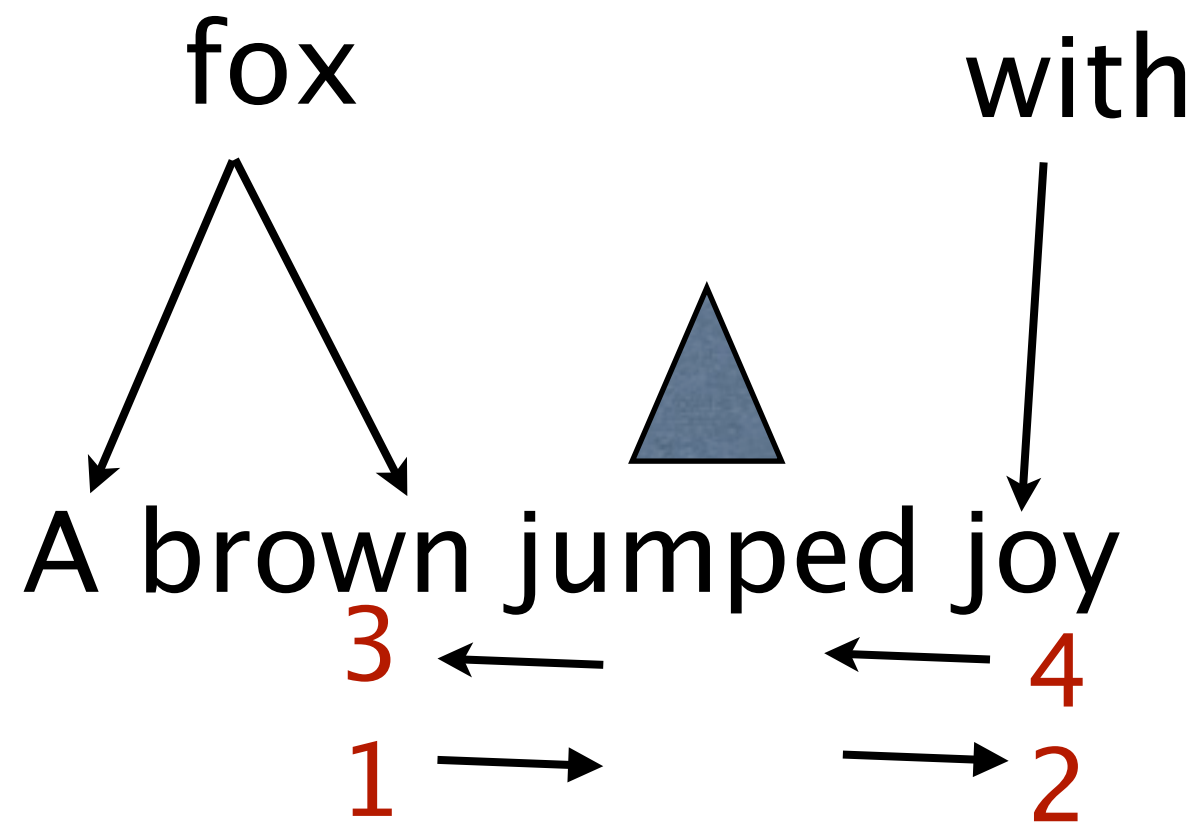
Example

Easy-First



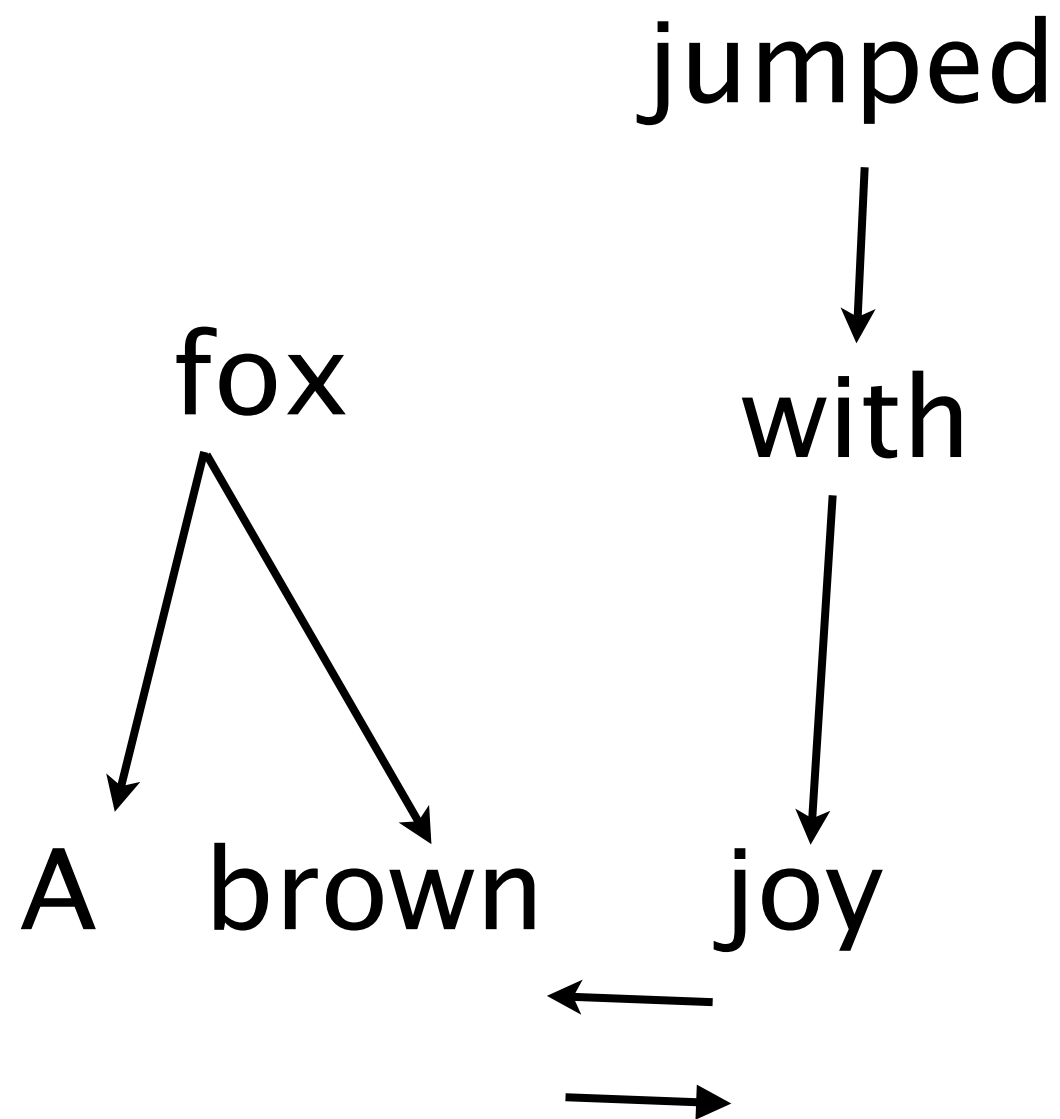
Example

Easy-First



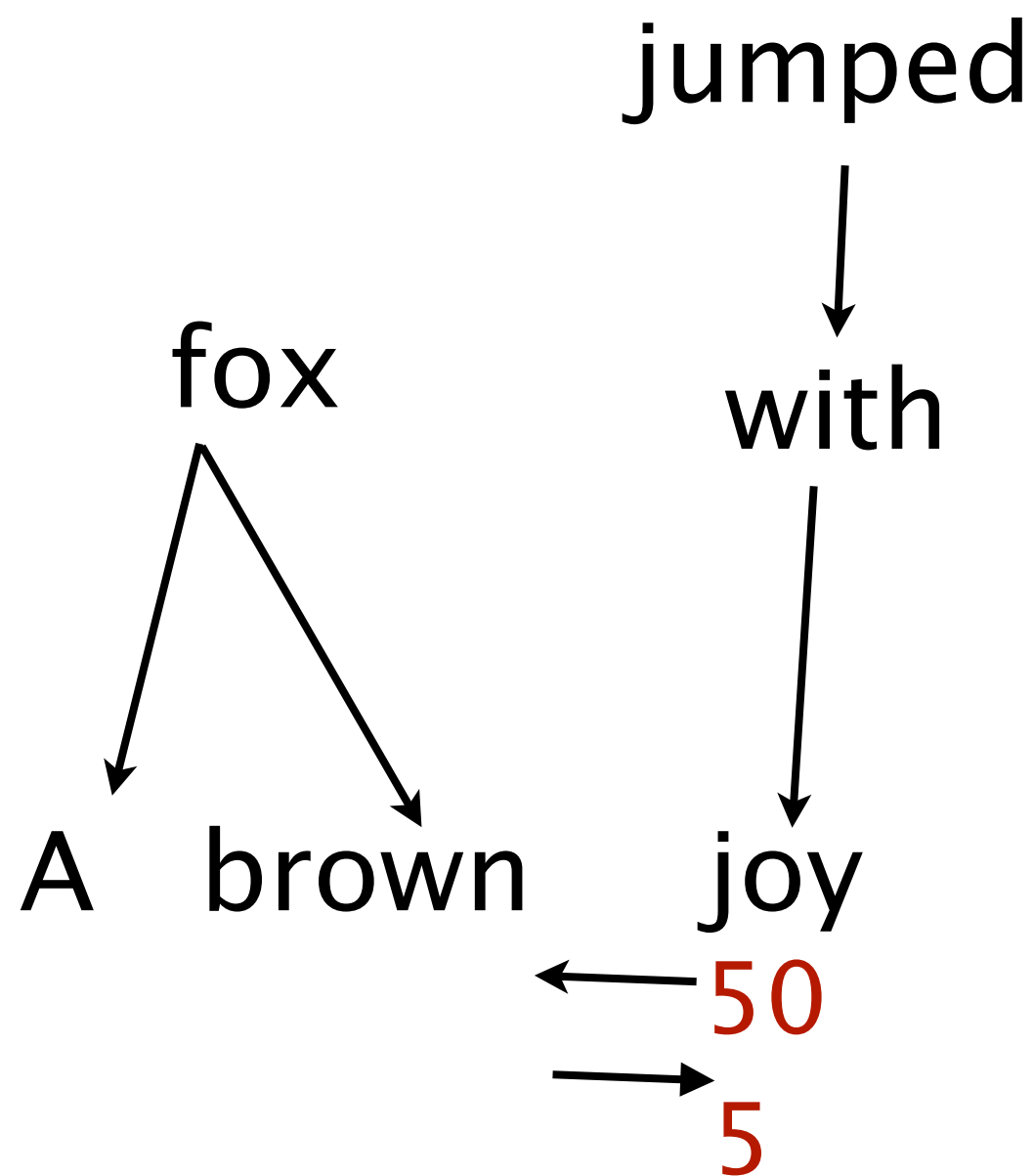
Example

Easy-First



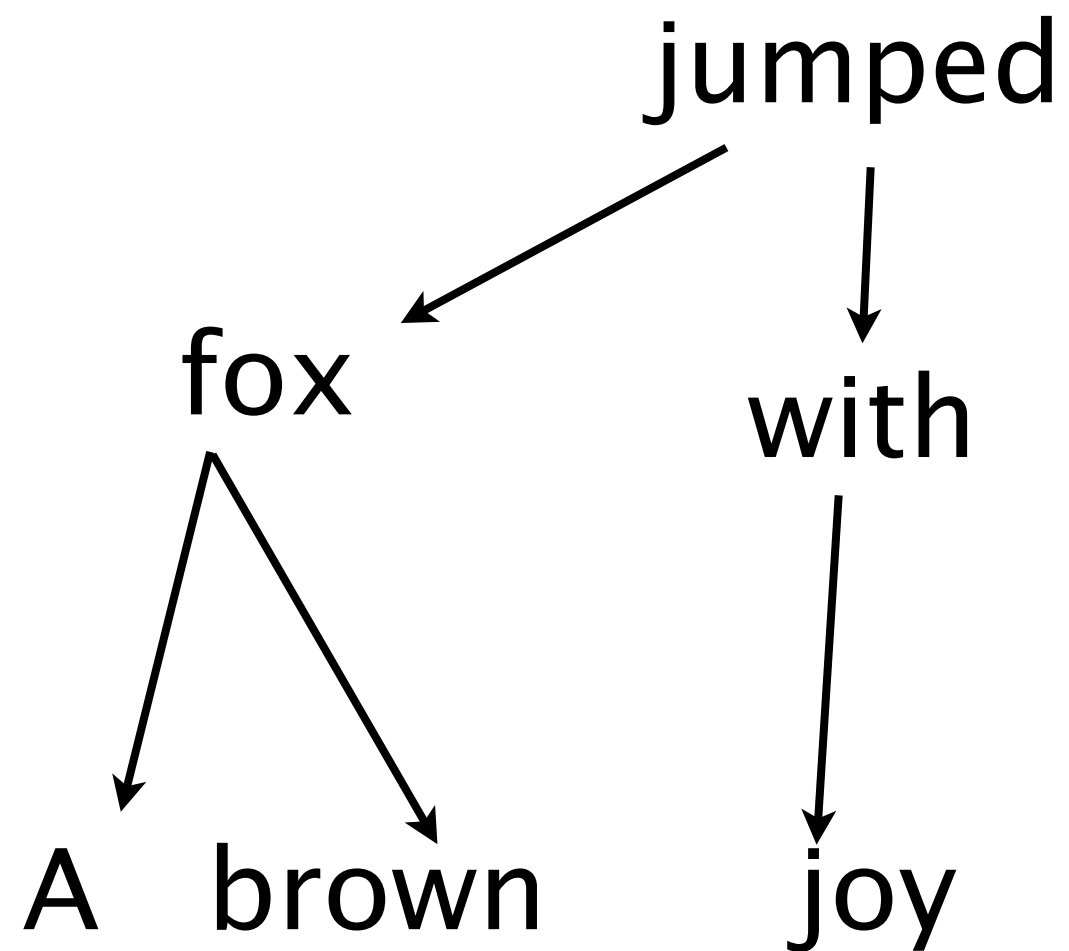
Example

Easy-First



Example

Easy-First



Learning

Easy-First

$$a^* = \arg \max_{\langle a' \rangle} w \phi(c, a')$$

Linear Classification

Perceptron

" c " for
context

" a' " for possible
attachments in context

Learning

Easy-First

$$a^* = \underset{\langle a' \rangle}{arg \max} w \phi(c, a')$$

Linear Classification

Perceptron

" c " for
context

" a' " for possible
attachments in context

Online Learning

- start with $w = 0$
- for each sentence:
 - while not done
 - calculate scores with w
 - if best action is **GOOD**
 - continue
 - Otherwise
 - call it a **BAD** action, decrease w
 - choose **GOOD** action, increase w

<http://www.cs.bgu.ac.il/~yoavg/publications/naacl2010dep.pdf>

Results (English)

	Unlabeled Accuracy	Root Accuracy	Exact Match
Malt	88.33	87.04	34.16
MST	90.05	93.95	34.64
Easy First	89.70	91.50	37.5

<http://www.cs.bgu.ac.il/~yoavg/publications/naacl2010dep.pdf>

Results (Hebrew)

	Gold Morphology	Predicted Morphology
Malt	80.07	73.4
MST	84.4	74.6
Easy First	84.2	76.2

<http://www.cs.bgu.ac.il/~yoavg/publications/mrlp2010dep.pdf>

Conclusions

- Dependency trees are a sound representation for MRLs
- Standard algorithms assume segmented and tagged input
- Improved inference can improve feature engineering (and accuracy)
- Even better algorithms for joint inference/learning are still needed

Think About It.

Think About It.





UPPSALA
UNIVERSITET

Thanks



מכון
ויצמן
למדע

reut.tsarfaty@weizmann.ac.il