Breaking down the Language Barrier with Statistical Machine Translation:
4) Optimization/Syntax for MT

http://www.phontron.com/class/sentan2014

Advanced Research Seminar I/III

Graham Neubig

2014-2-6

Assignment

- (Only one assignment this week)
- You are given a baseline machine translation system
  - LM/Alignment: Baseline from exercises 1, 2
  - TM: Phrases of up to length 4
  - SM: Uniform distribution
  - RM: Distortion penalty
  - Reordering Limit: 6
- Try to improve its accuracy by changing one of the features listed above, or anything else

Probabilistic Model for Translation

Formal Definition of Translation

- A translation is defined as (in opposite order)
  - Output sentence E
  - Derivation D
  - Input sentence F

E = hello where is the station

D =

D_{ep} =	hello	where is	the station
D_{fp} =	こんにちは	どこ です か	駅 は
D_{o} =	0	2	1

F = こんにちは 駅 は どこ です か

Probabilistic Modeling of Translation

- We want a probability of D and E given F: $P(D, E|F)$
- Use Bayes's law and note that P(F) doesn't affect results

$$P(D, E|F) = P(D, E, F) / P(F)$$
$$\propto P(D, E, F)$$

- And split the probabilities further

$$P(D, E, F) \propto P(E) * P(D_{ep}|E) * P(D_{fp}|D_{ep}, E) * P(D_{order}|D_{fp}, D_{ep}, E) * P(F|D_{order}, D_{fp}, D_{ep}, E)$$

Language Model

Segmentation Model

Translation Model

Reordering Model

Always P=1 (F is decided by D)

Log-Linear Combination

- We can also calculate log probability $P(D, E|F)$

$$\log P(D, E, F) \propto \log P(E) + \quad \text{Language Model}$$
$$\log P(D_{ep}|E) + \quad \text{Segmentation Model}$$
$$\log P(D_{fp}|D_{ep}, E) + \quad \text{Translation Model}$$
$$\log P(D_{order}|D_{fp}, D_{ep}, E) \quad \text{Reordering Model}$$

- And generalize this as combination of features

$$\log P(D, E, F) \propto \varphi_{LM}(D, E, F) +$$
$$\varphi_{SM}(D, E, F) +$$
$$\varphi_{TM}(D, E, F) +$$
$$\varphi_{RM}(D, E, F)$$

Why Features?

- Scores of translation, reordering, and language models

	LM	TM	RM	
○ Taro visited Hanako	-4	-3	-1	-8
X the Taro visited the Hanako	-5	-4	-1	-10
X Hanako visited Taro	-2	-3	-2	-7 ← Best Score X

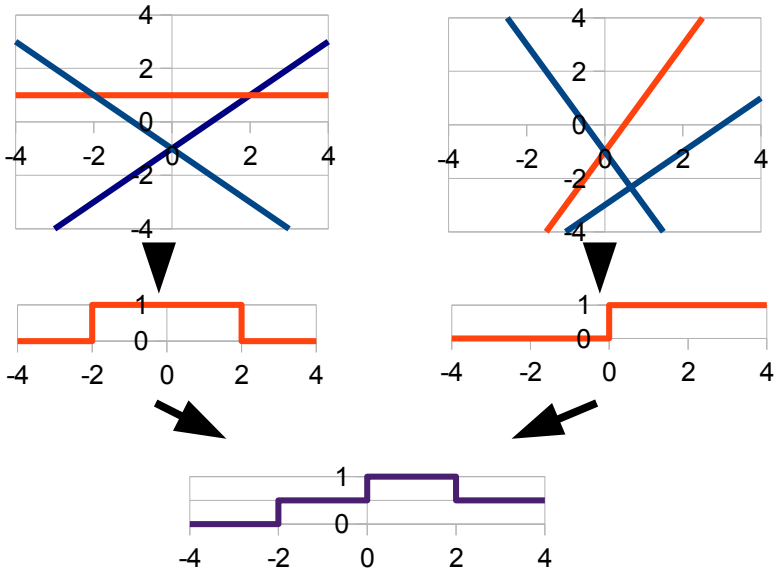
- If we add weights, we can get better answers:

	LM	TM	RM	
○ Taro visited Hanako	0.2*-4	0.3*-3	0.5*-1	-2.2 ← Best Score ○
X the Taro visited the Hanako	0.2*-5	0.3*-4	0.5*-1	-2.7
X Hanako visited Taro	0.2*-2	0.3*-3	0.5*-2	-2.3

- Optimization finds these weights: $w_{LM}=0.2$ $w_{TM}=0.3$ $w_{RM}=0.5$

4 Major Techniques in Optimization

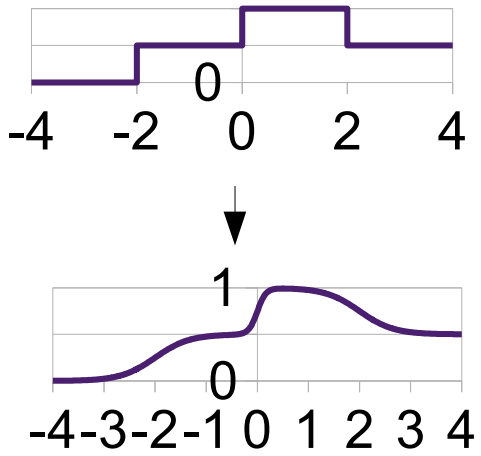
MERT [Och+ 04]



Online (MIRA) [Watanabe+ 07]

$$\boldsymbol{w}_t = \boldsymbol{w}_{t-1} + \varphi\left(e_i^*\right) - \varphi\left(\hat{e}_i\right)$$

Gradient Based (xBLEU) [Smith+ 06]



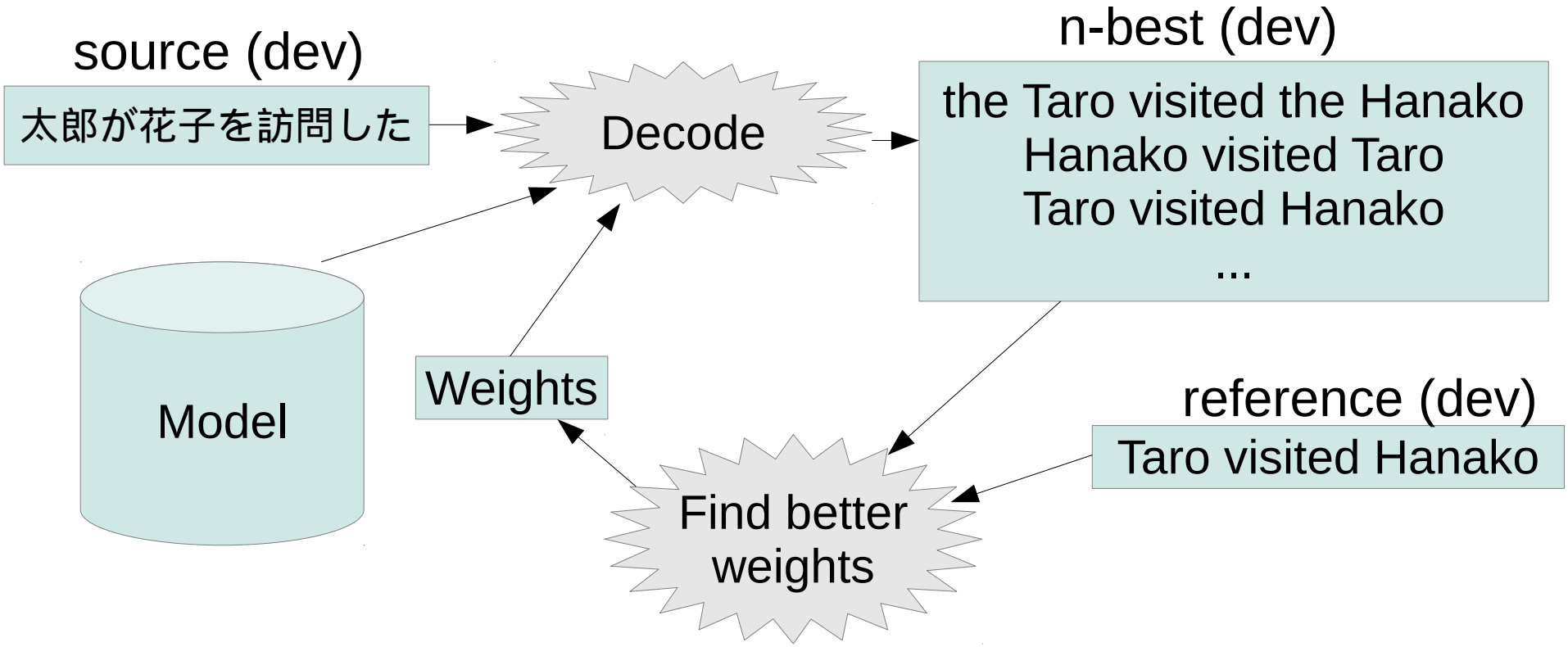
PRO [Hopkins+ 11]

e1,1: #2
e1,2: #1
e1,3: #3

Minimum Error Rate Training (MERT)

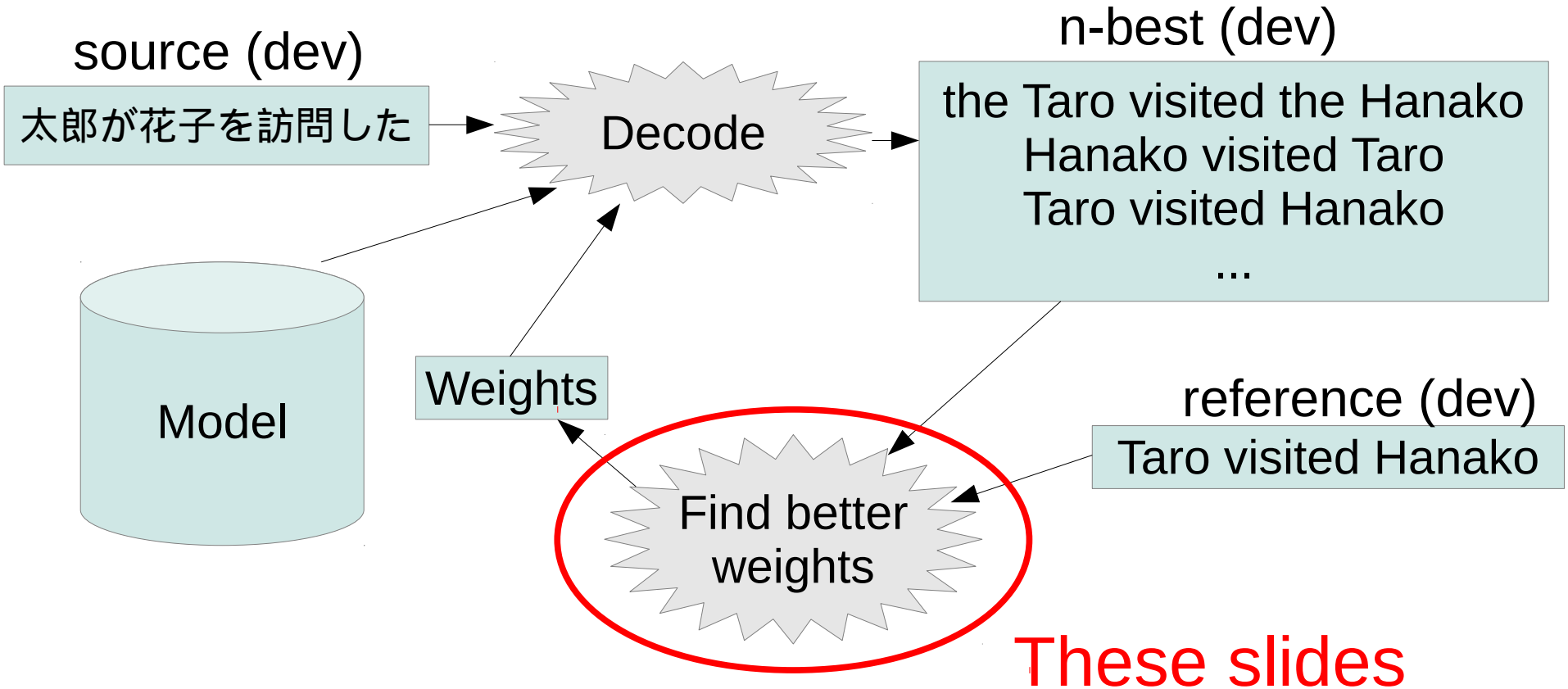
Minimum Error Rate Training (MERT)

- MERT performs iterations to increase the score [Och 03]



MERT

- MERT performs iterations to increase the score [Och 03]



MERT Weight Update:

- Adjust one weight at a time

	Weights			Score
	w_{LM}	w_{TM}	w_{RM}	
Initial:	0.1	0.1	0.1	0.20
Optimize w_{LM}:				
	0.4	0.1	0.1	0.32
Optimize w_{TM}:				
	0.4	0.1	0.1	0.32
Optimize w_{RM}:				
	0.4	0.1	0.3	0.4
Optimize w_{LM}:				
	0.35	0.1	0.3	0.41
Optimize w_{TM}:				

Updating One Weight:

- We start with:
  n-best list

f_1	ϕ_{LM}	ϕ_{TM}	ϕ_{RM}	BLEU*
$e_{1,1}$	1	0	-1	0
$e_{1,2}$	0	1	0	1
$e_{1,3}$	1	0	1	0

f_2	ϕ_{LM}	ϕ_{TM}	ϕ_{RM}	BLEU*
$e_{2,1}$	1	0	-2	0
$e_{2,2}$	3	0	1	0
$e_{2,3}$	2	1	2	1

fixed weights:

$$w_{LM}=-1,\ w_{TM}=1$$

weight to be adjusted:

$$w_{RM}=???$$

* Calculating BLEU for one sentence is a bit simplified, usually we compute for the whole corpus

Updating One Weight:

- Next, transform each hypothesis into lines:

$$y = a\,x + b$$

- Where:
  - a is the value of the feature to be adjusted
  - b is the weighted sum of the fixed features
  - x is the weight to be adjusted (unknown)

Updating One Weight:

- Example:

w_{LM}=-1, w_{TM}=1, w_{RM}=???

$$y = a\,x + b$$

$$a = \varphi_{RM} \qquad b = w_{LM}\,\varphi_{LM} + w_{TM}\,\varphi_{TM}$$

f_1	ϕ_{LM}	ϕ_{TM}	ϕ_{RM}
$e_{1,1}$	1	0	-1
$e_{1,2}$	0	1	0
$e_{1,3}$	1	0	1

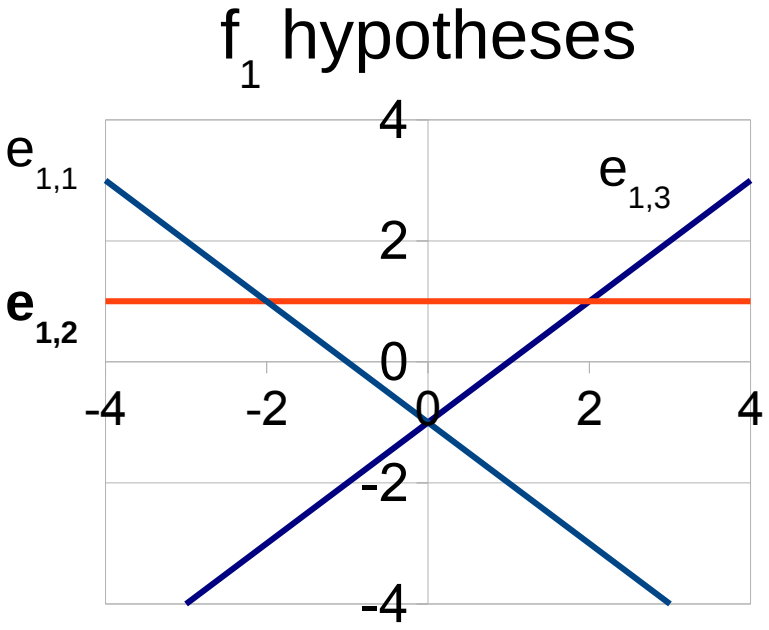
$a_{1,1}$=-1 $b_{1,1}$=-1

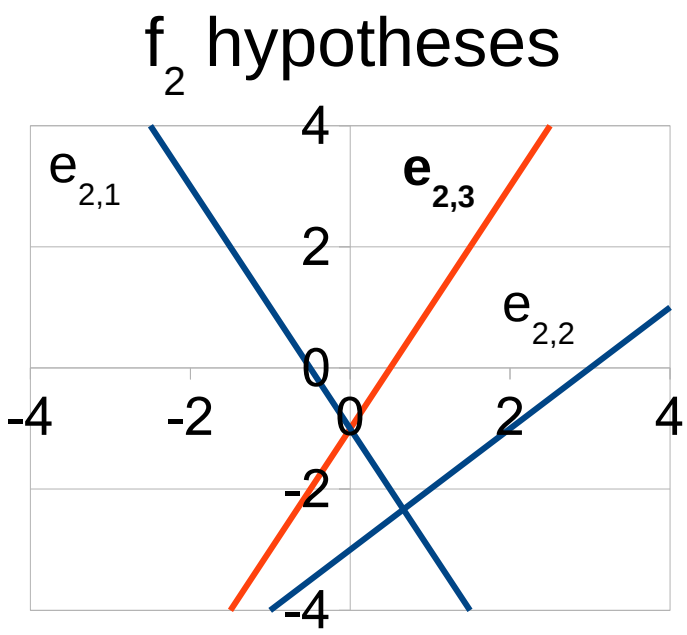
$a_{1,2}$=0 $b_{1,2}$=1

$a_{1,3}$=1 $b_{1,3}$=-1

Updating One Weight:

- Draw lines on a graph: $y = a\,x + b$



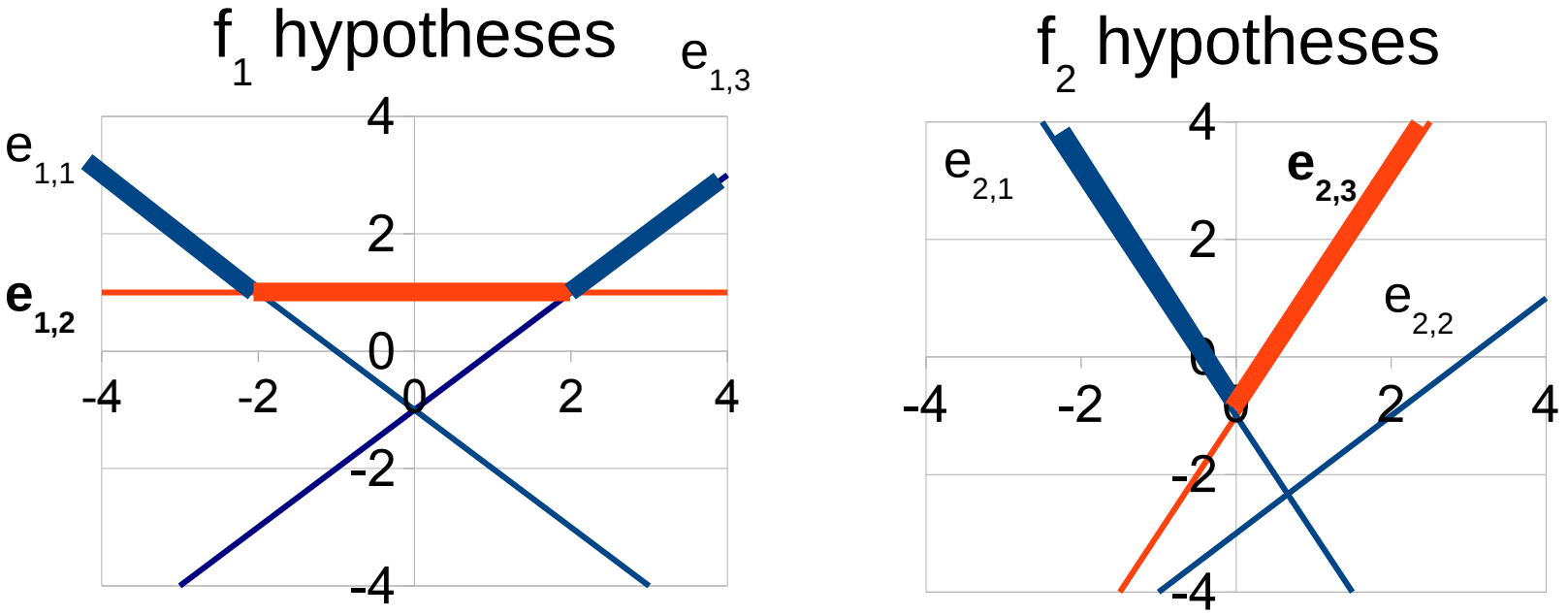


$a_{1,1}$=-1 $b_{1,1}$=-1
$a_{1,2}$=0 $b_{1,2}$=1
$a_{1,3}$=1 $b_{1,3}$=-1

$a_{2,1}$=-2 $b_{2,1}$=-1
$a_{2,2}$=1 $b_{2,2}$=-3
$a_{2,3}$=-2 $b_{2,3}$=1

Updating One Weight:

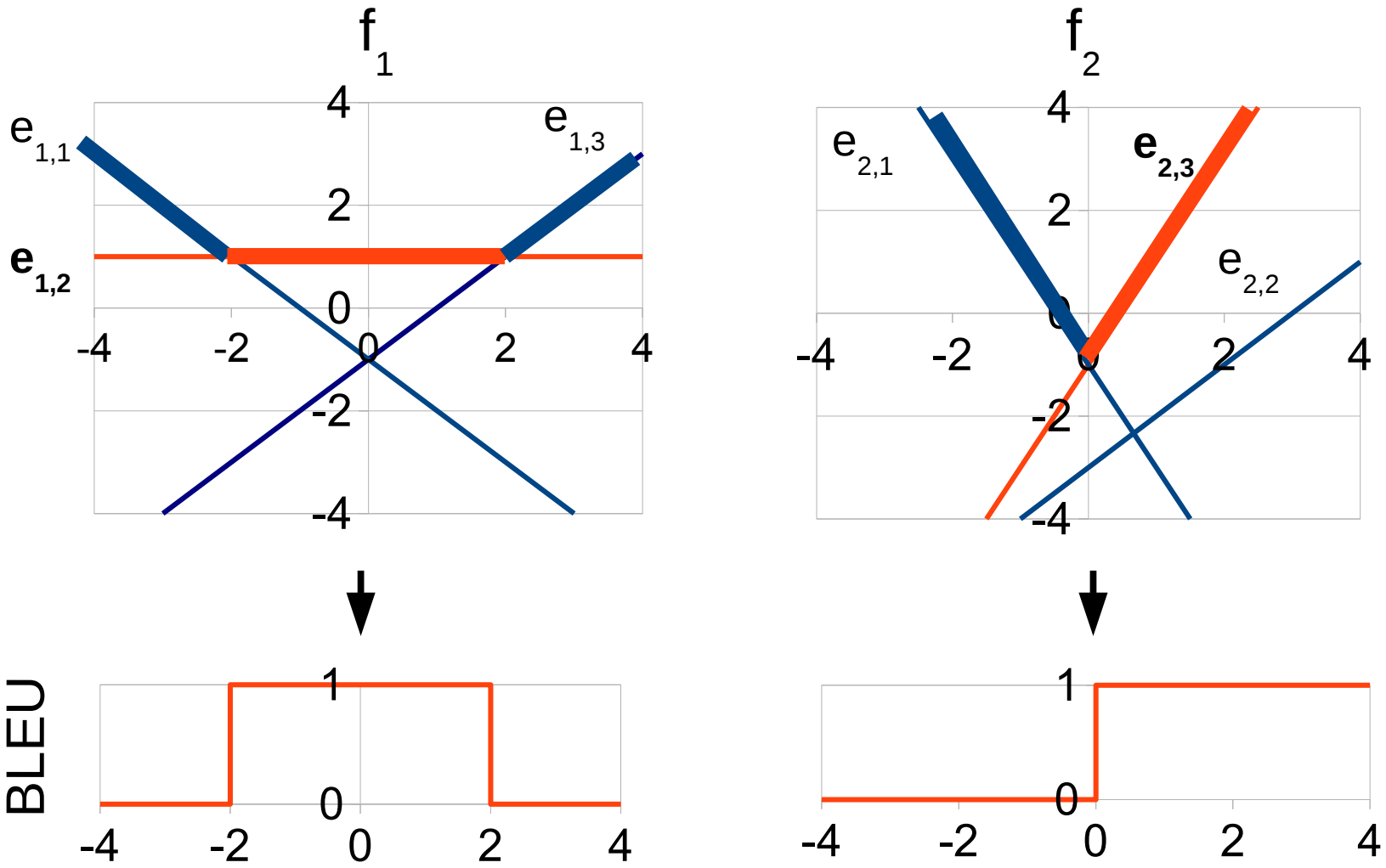
- Find the lines that are highest for each range of *x*:



- This is called the convex hull (or upper envelope)

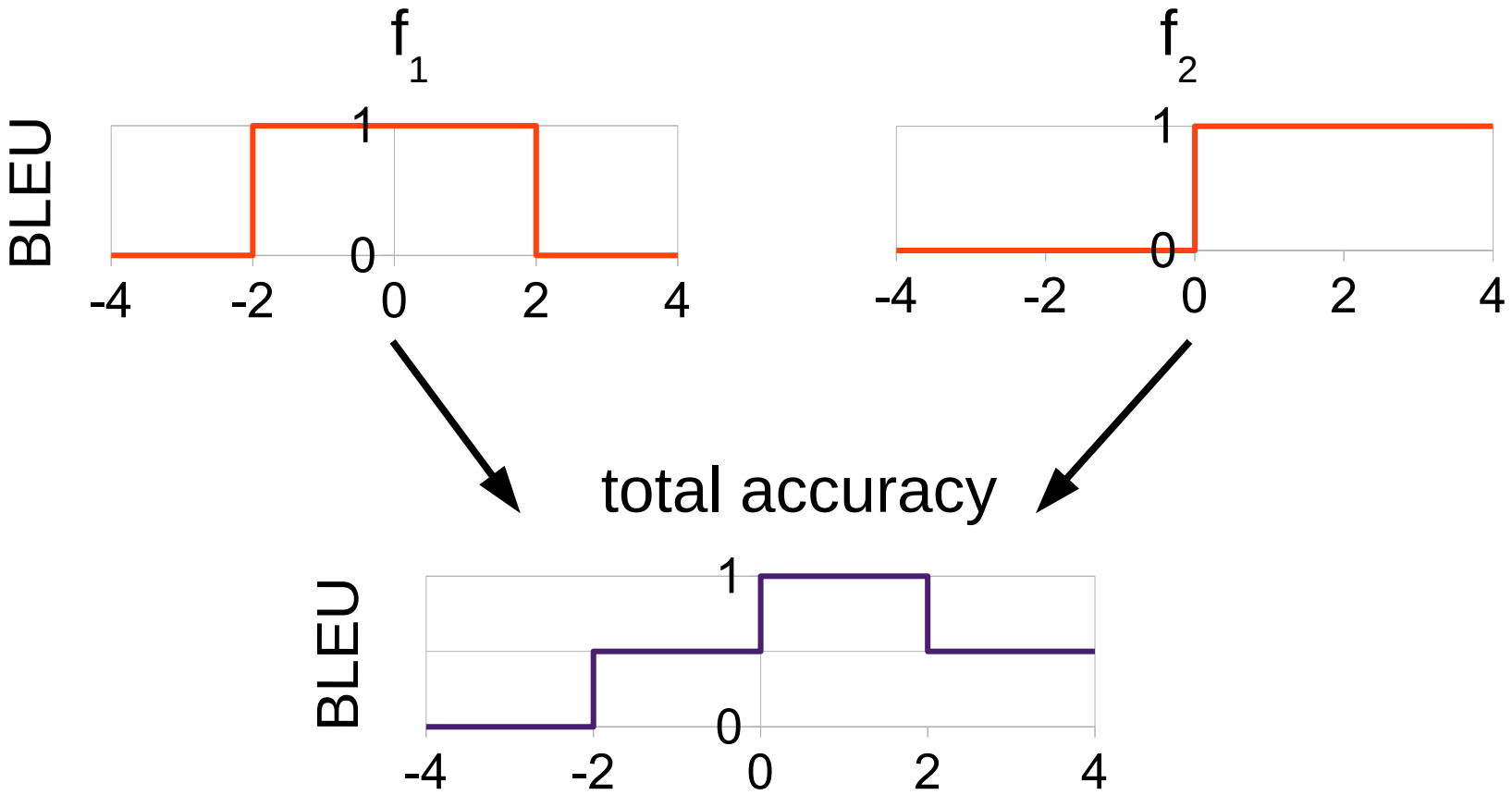
Updating One Weight:

- Using the convex hull, find scores at each range:



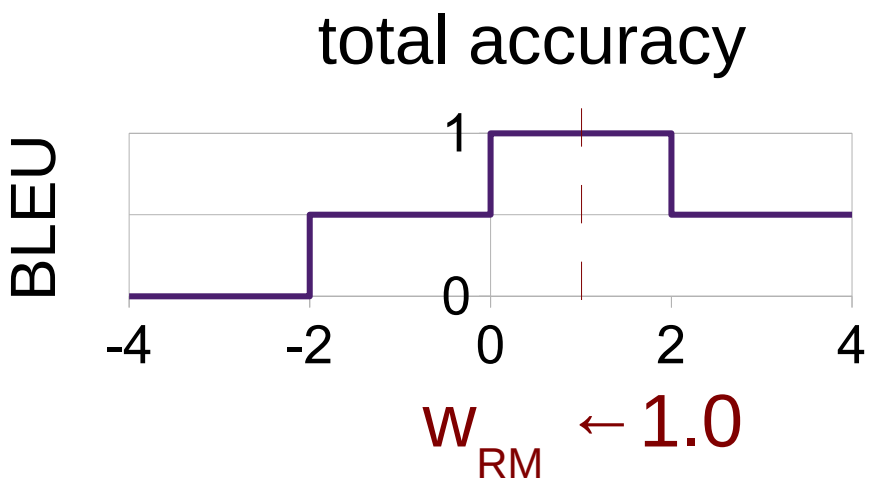
Updating One Weight:

- Combine multiple sentences into a single error plane:



Updating One Weight:

- Choose middle of best region:



Summary

- For each sentence:
  - Create lines for each n-best hypothesis
  - Combine lines and find upper envelope
  - Transform upper envelope into error surface
- Combine error surfaces into one
- Find the range with the highest score
- Set the weight to the middle of the range

Standard Features for MT

Log-Linear Combination

- Our first combination is motivated by the standard model

$$w_{LM}\varphi_{LM}(D,E,F)+w_{SM}\varphi_{SM}(D,E,F)+$$
$$w_{TM}\varphi_{TM}(D,E,F)+w_{RM}\varphi_{RM}(D,E,F)$$

- But actually, if it raises accuracy, we can add any other features we want!

$$w_{LM}\varphi_{LM}(D,E,F)+w_{SM}\varphi_{SM}(D,E,F)+$$
$$w_{TM}\varphi_{TM}(D,E,F)+w_{RM}\varphi_{RM}(D,E,F)+\ldots$$

Word Penalty

- One of the most useful features is “word penalty” (actually name is confusing, penalty/bonus both OK)
- This feature gets a value equal to the sentence length

E = hello where is the station

$$\varphi_{\mathrm{WP}}(D, E, F) = |E|$$

- If we set w_{WP} higher, we get longer sentences
- If we set w_{WP} lower, we get shorter sentences
- This is important for BLEU, which likes sentences the same length as the reference

More Translation Probabilities

- In our traditional model, we only use

$$P(D_{fp}|D_{ep}) = \prod_{k=1}^{K} P(fp_k|ep_k)$$

- But we can also calculate target given source

$$P(D_{ep}|D_{fp}) = \prod_{k=1}^{K} P(ep_k|fp_k)$$

- Also, "lexical translation probabilities" using Model 1

$$P_{lex}(D_{fp}|D_{ep}) = \prod_{k=1}^{K} \prod_{i=1}^{|ep_k|} \frac{1}{|fp_k|+1} \sum_{j=1}^{|fp_k|+1} P(ep_{k,i}|fp_{k,j})$$

Model one

Word-based, helps with sparsity!

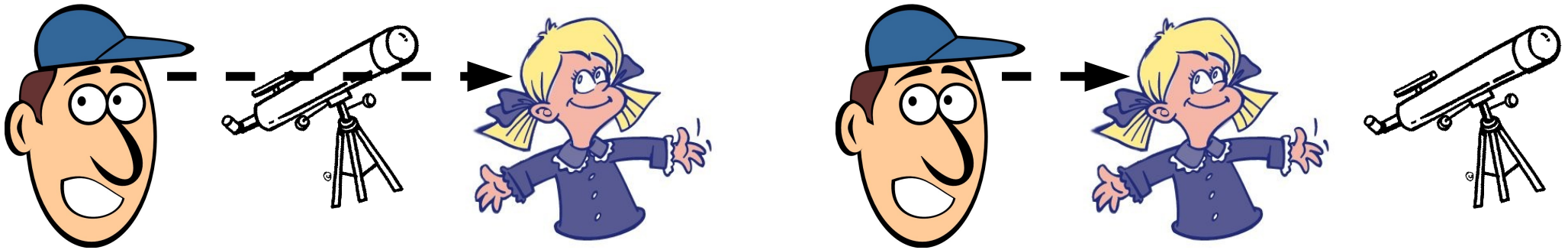
Many Others!

- Add multiple reordering models!
- Add multiple language models!
- Add a penalty when a parts of speech are different for different languages!
- ...

What is Syntax?

Interpreting Language is Hard!

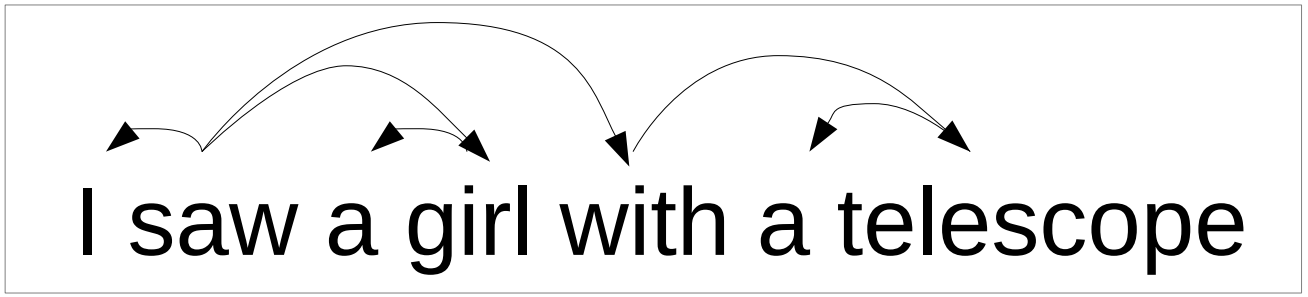
I saw a girl with a telescope



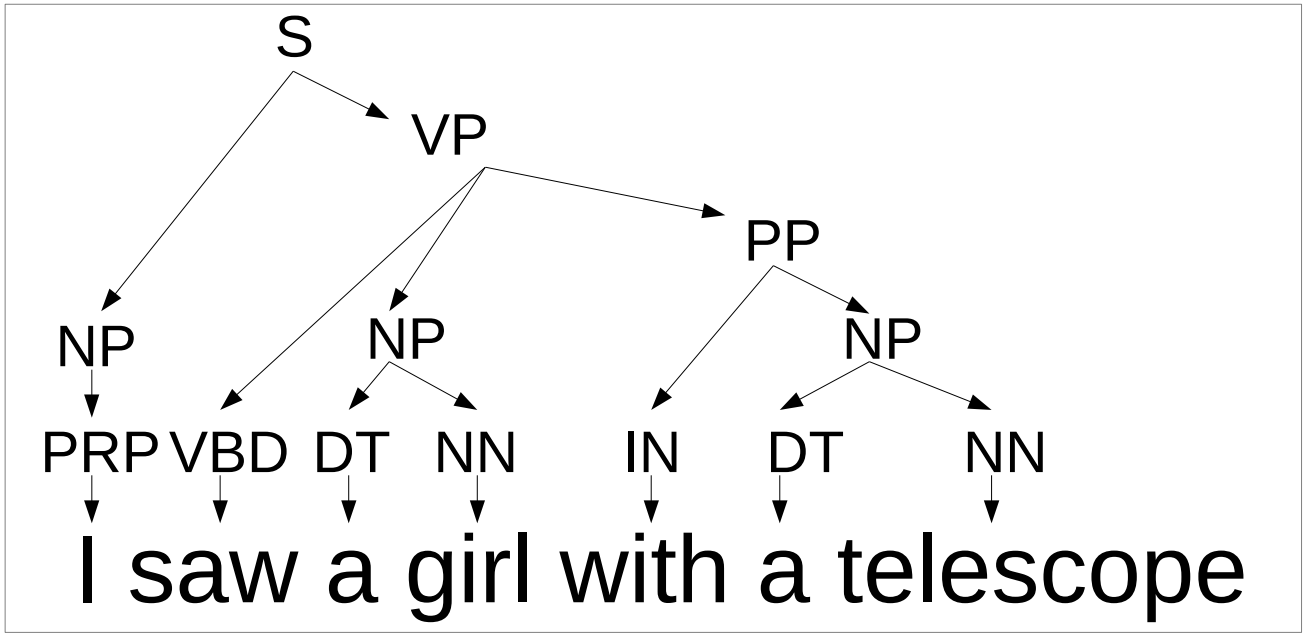
- “Parsing” resolves structural ambiguity in a formal way

Two Types of Syntactic Structure

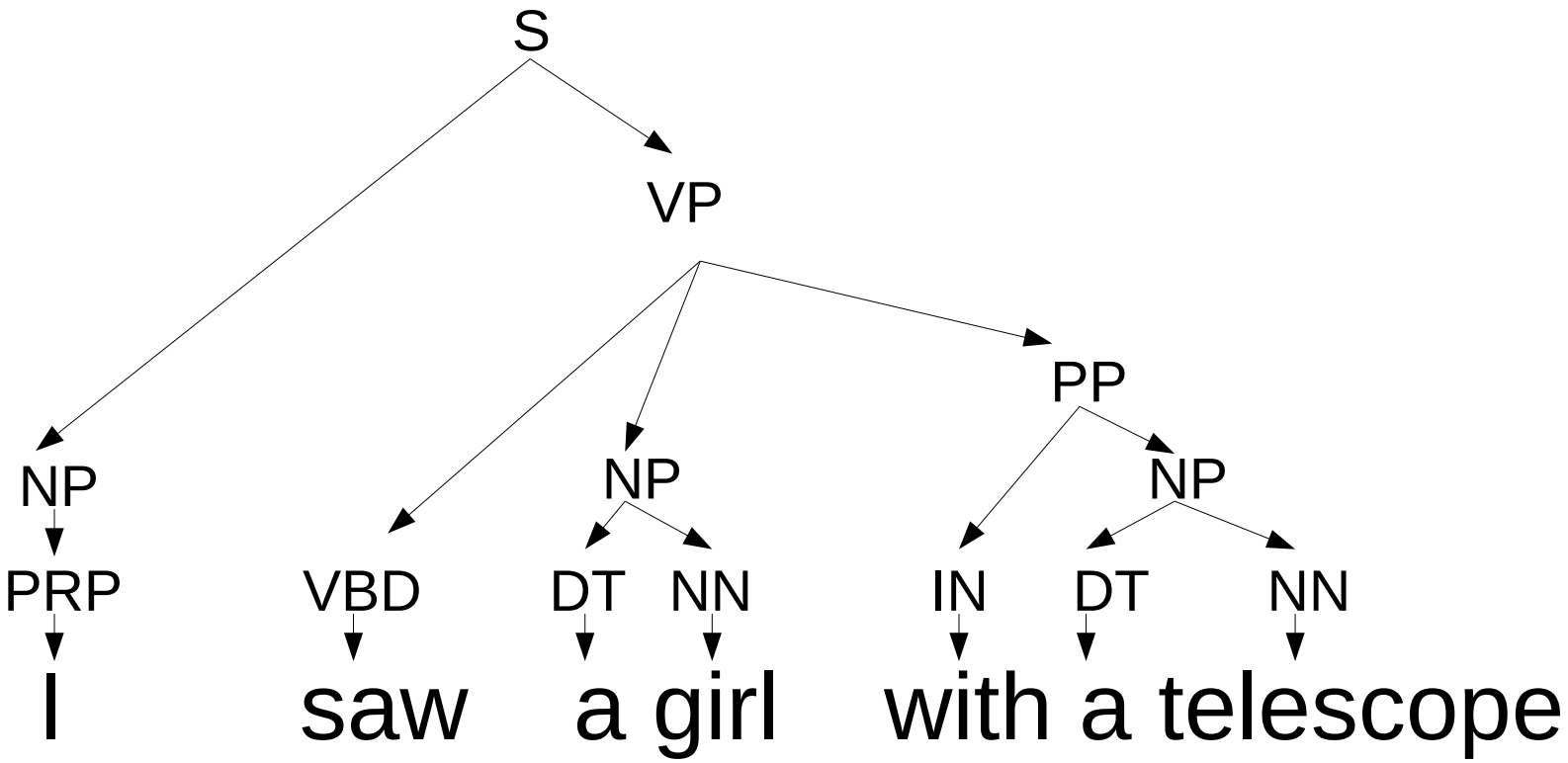
- Dependency: focuses on relations between words



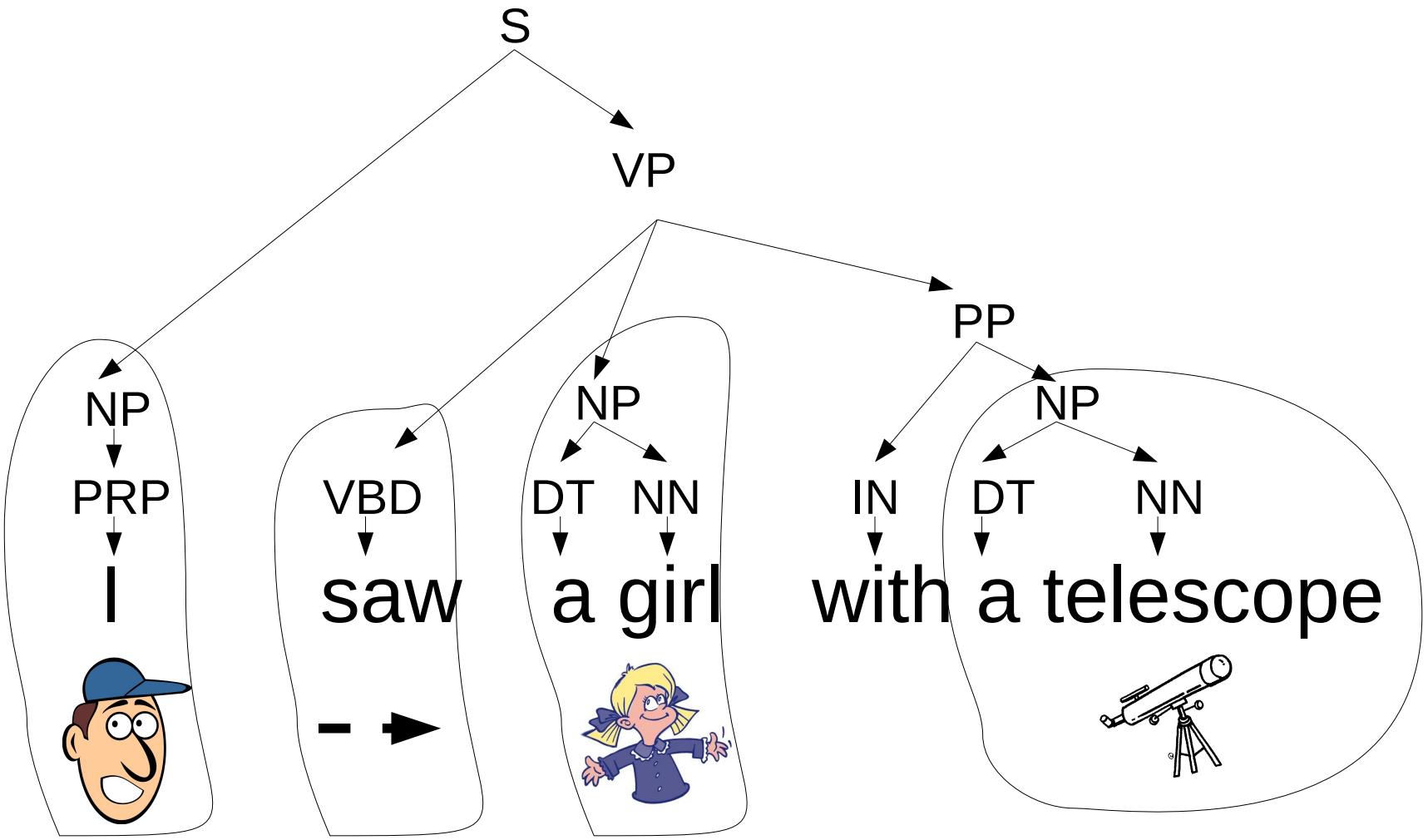
- Phrase structure: focuses on identifying phrases and their recursive structure



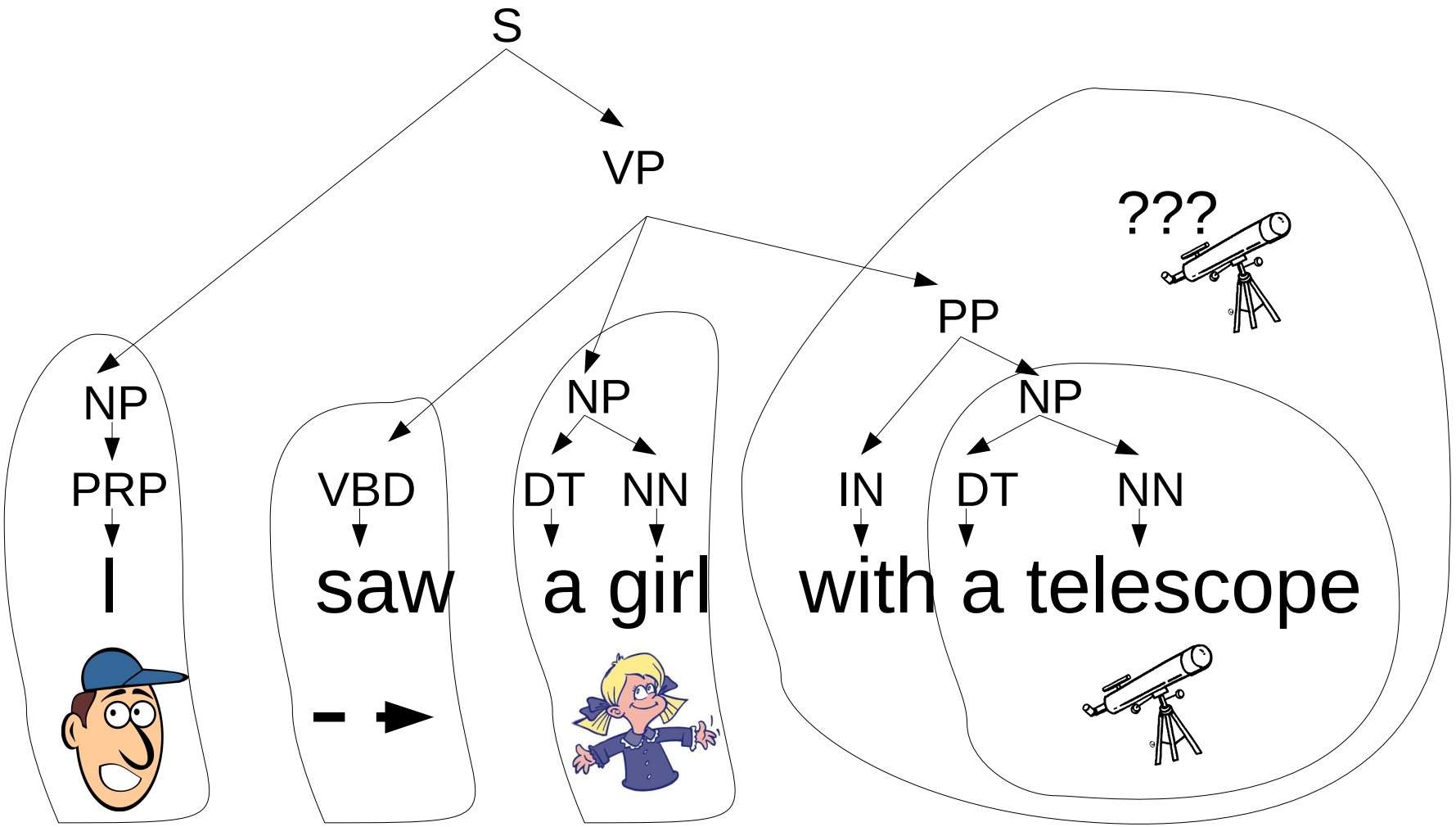
Recursive Structure?



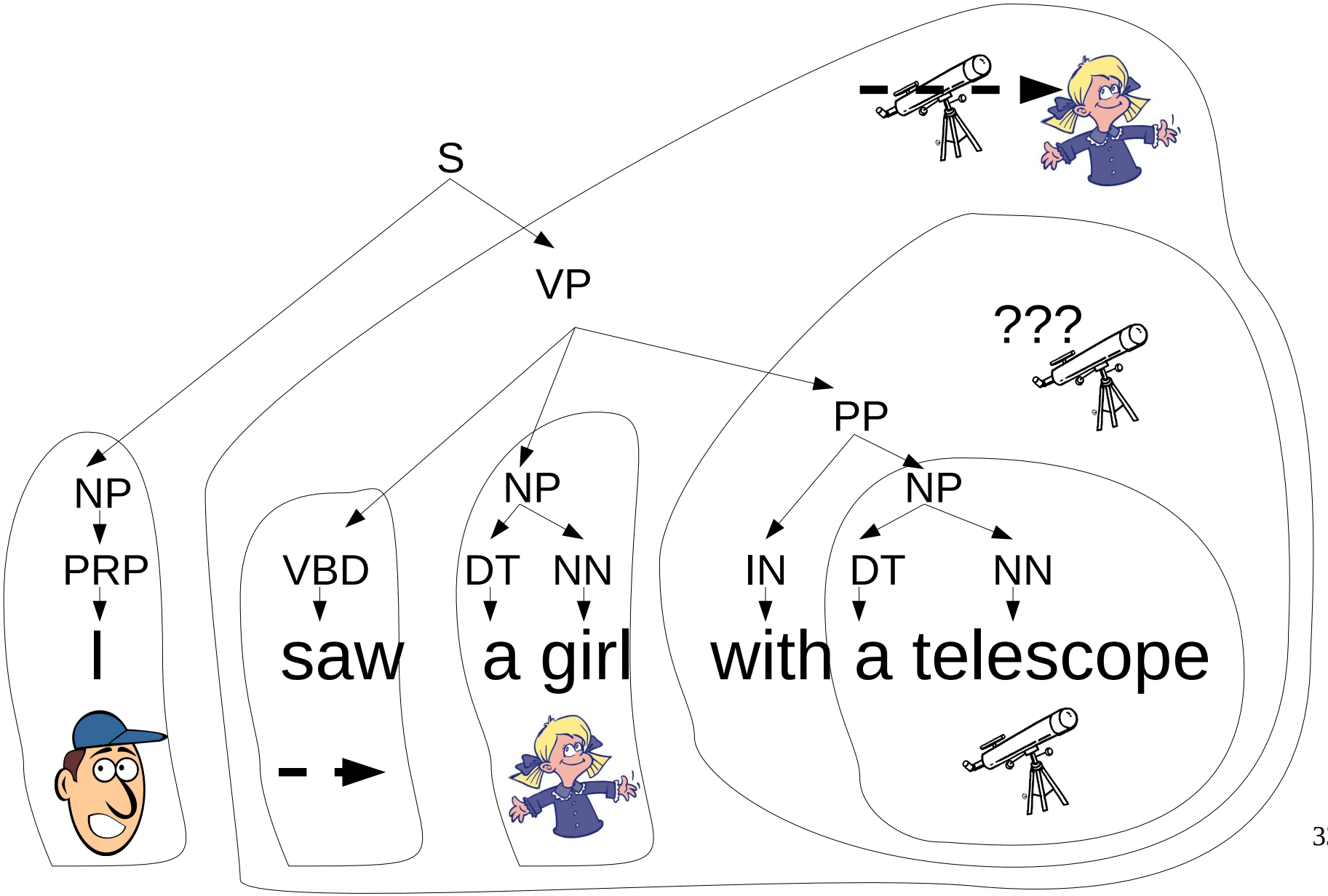
Recursive Structure?



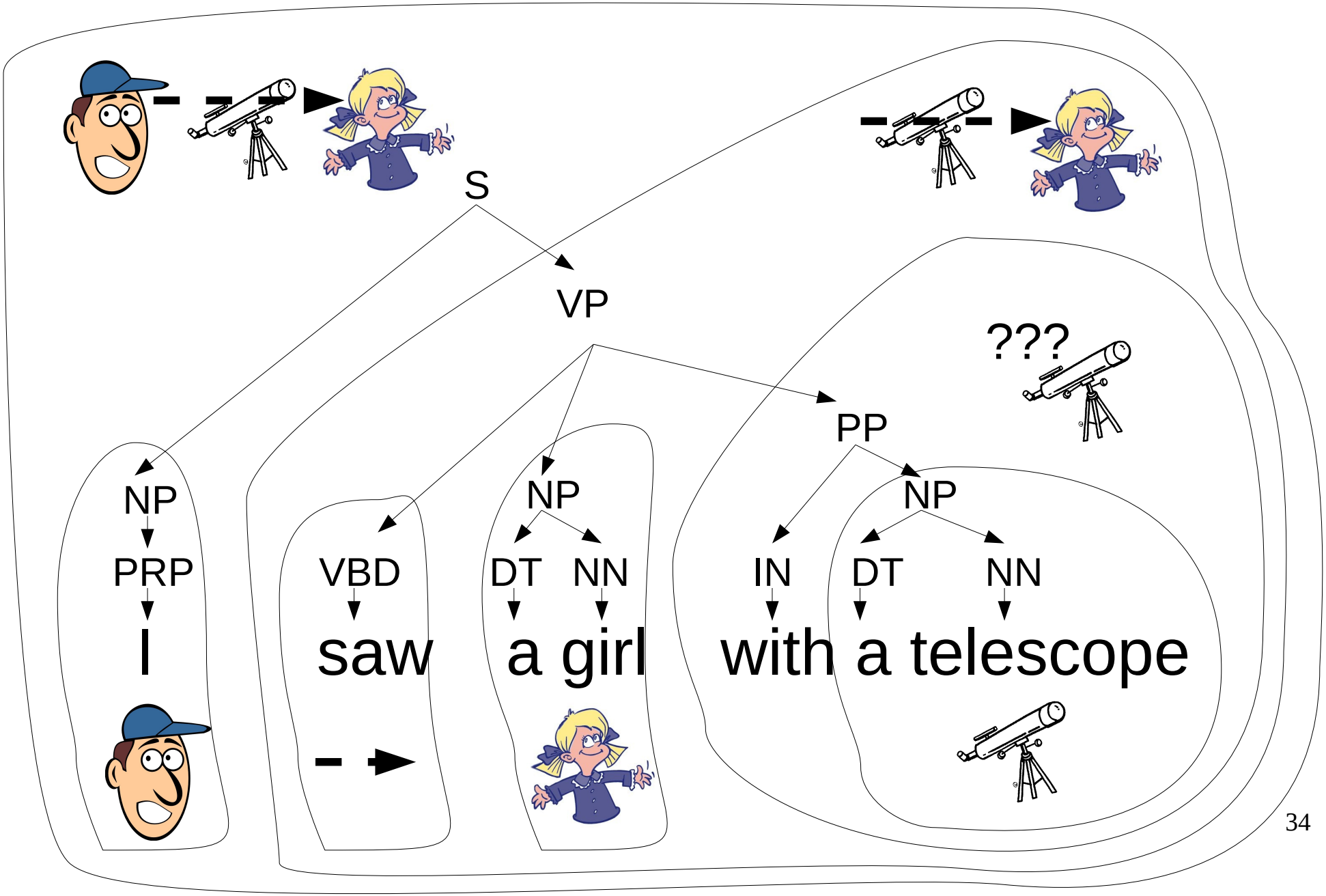
Recursive Structure?



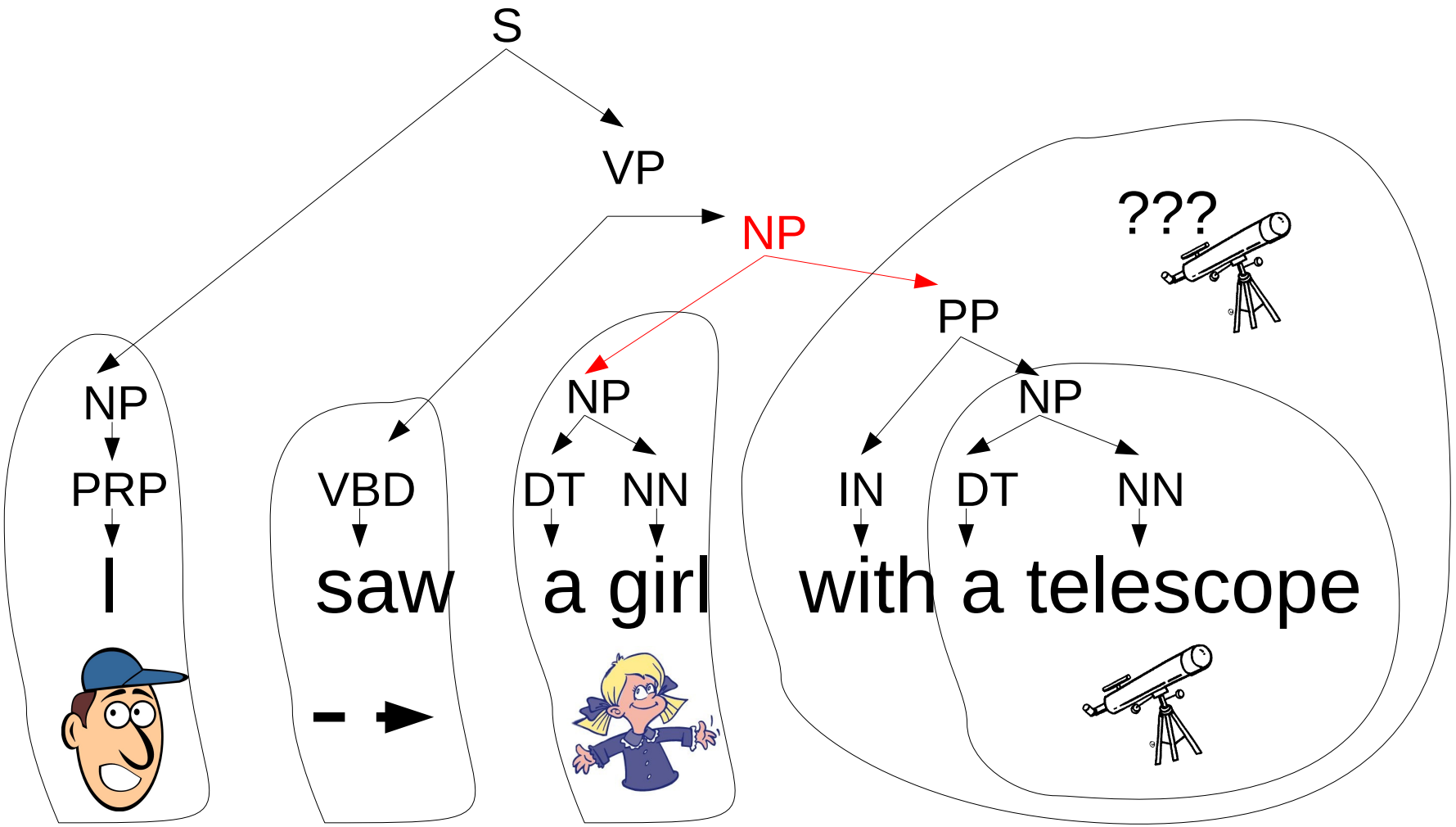
Recursive Structure?



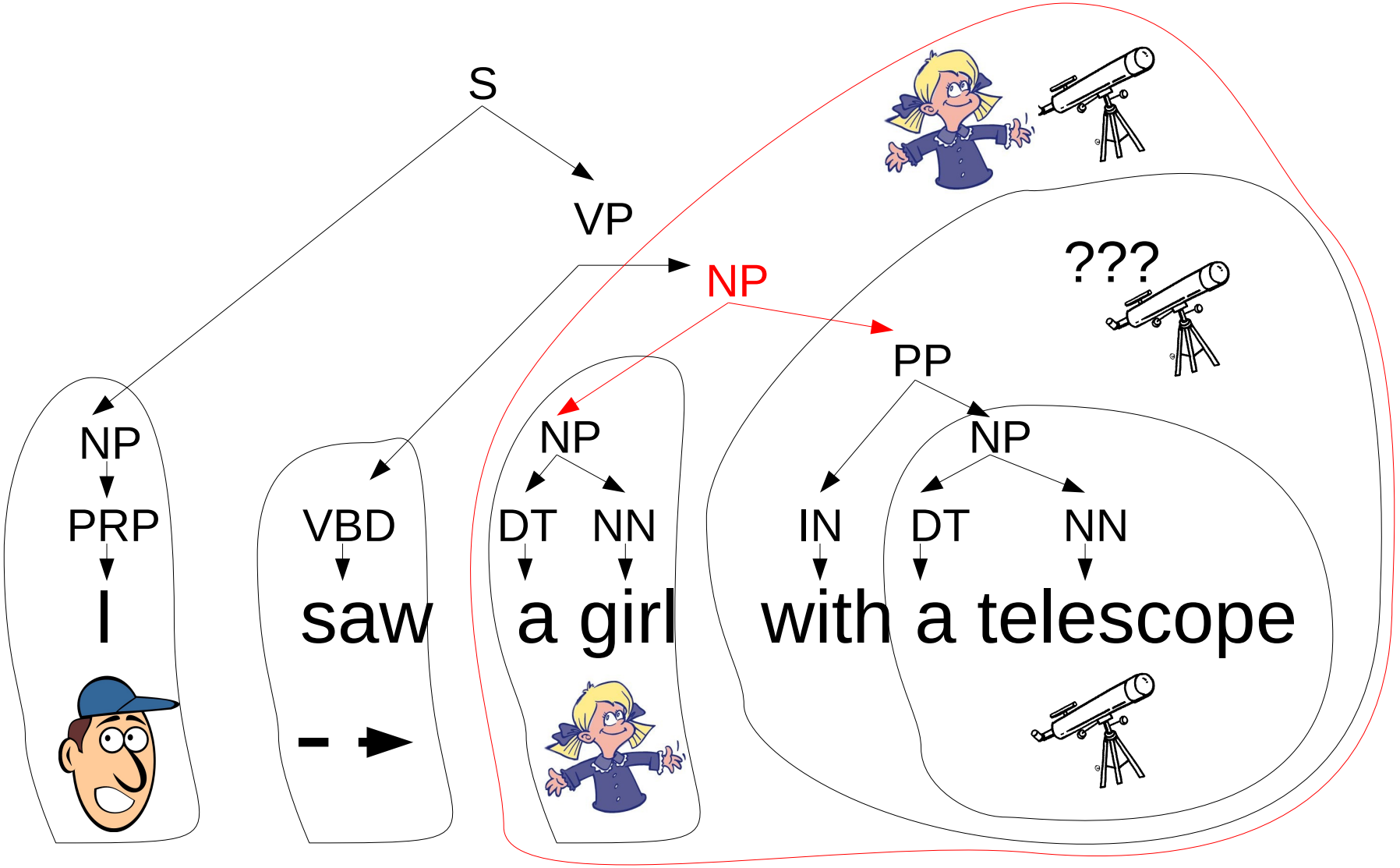
Recursive Structure?



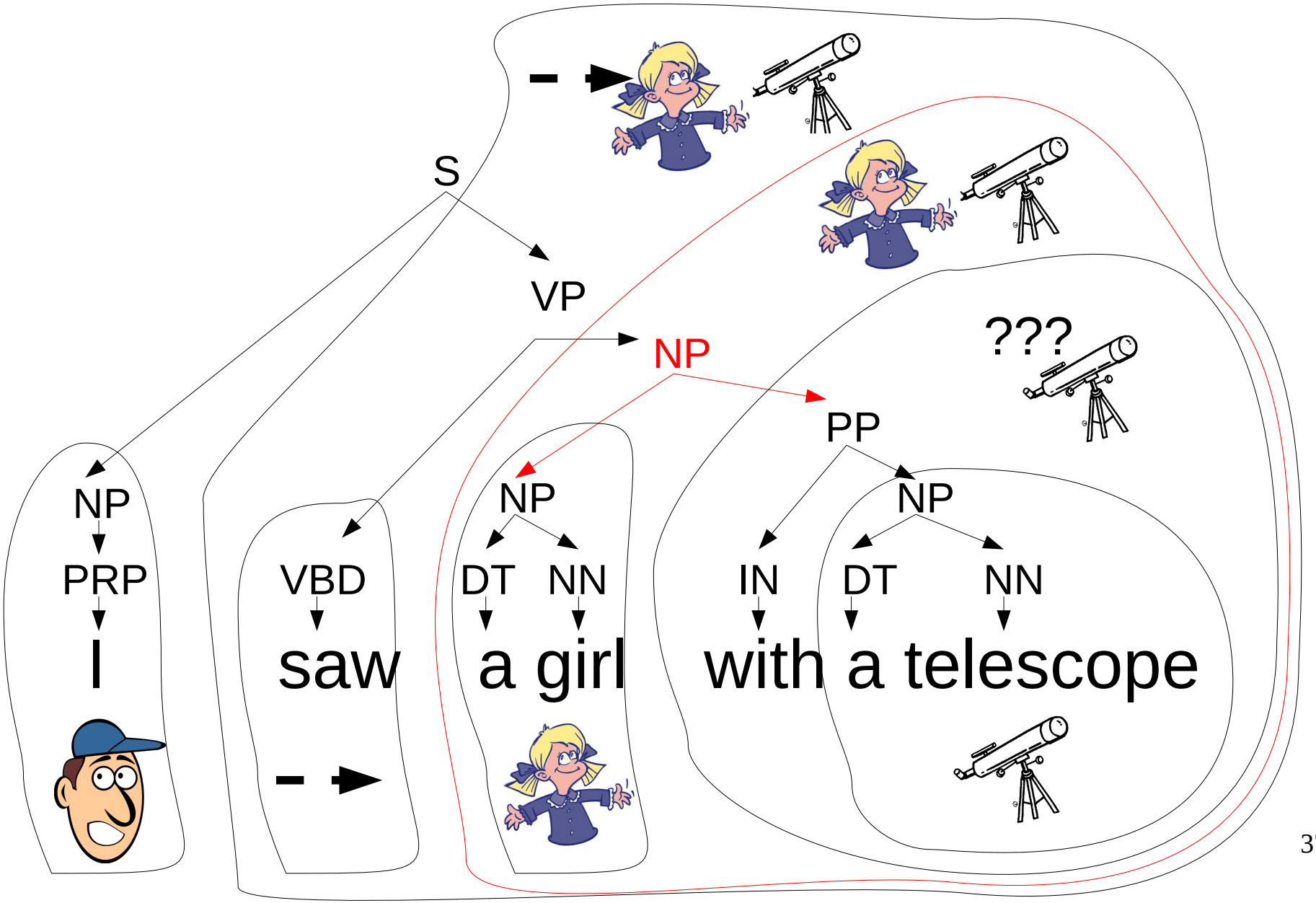
Different Structure, Different Interpretation



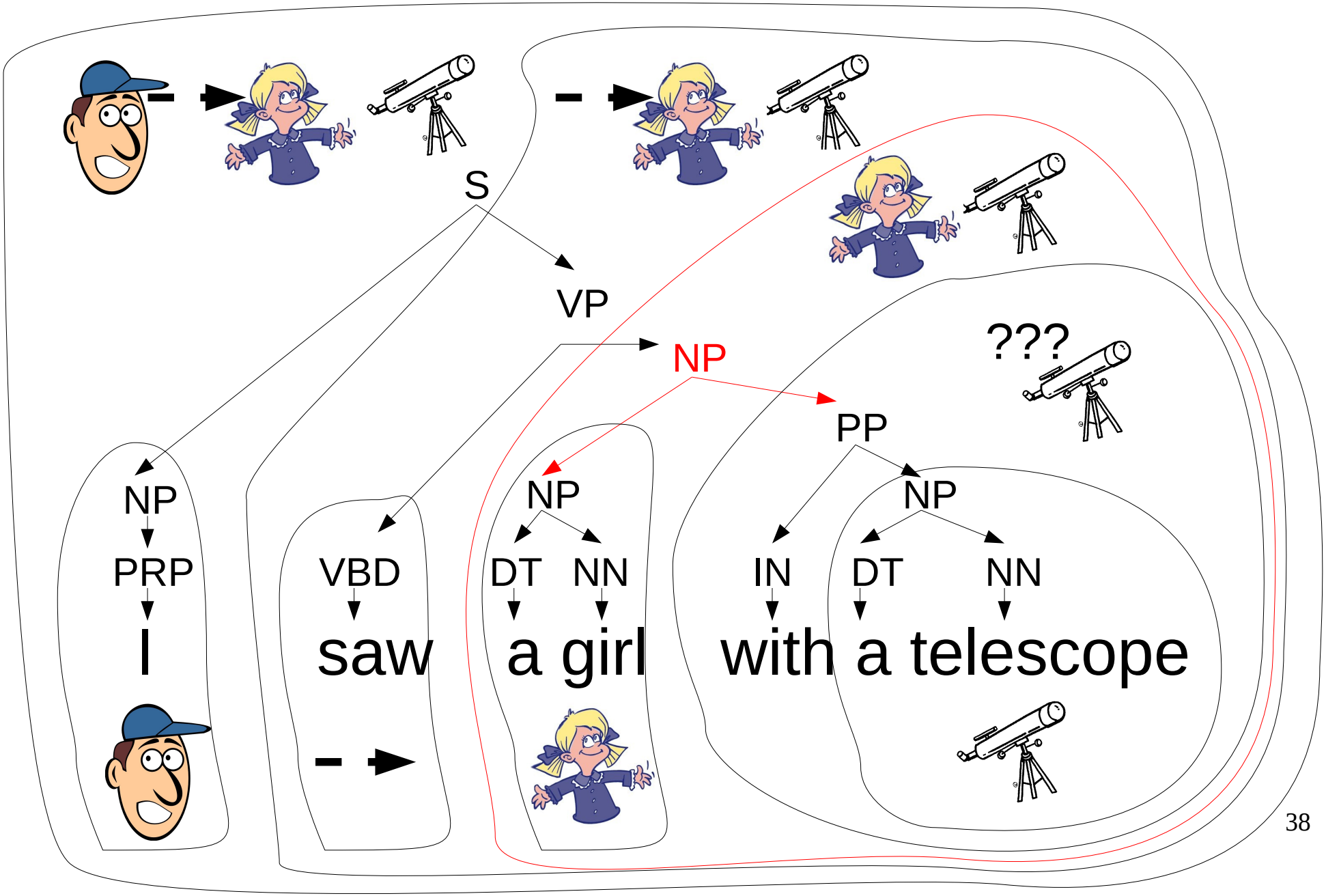
Different Structure, Different Interpretation



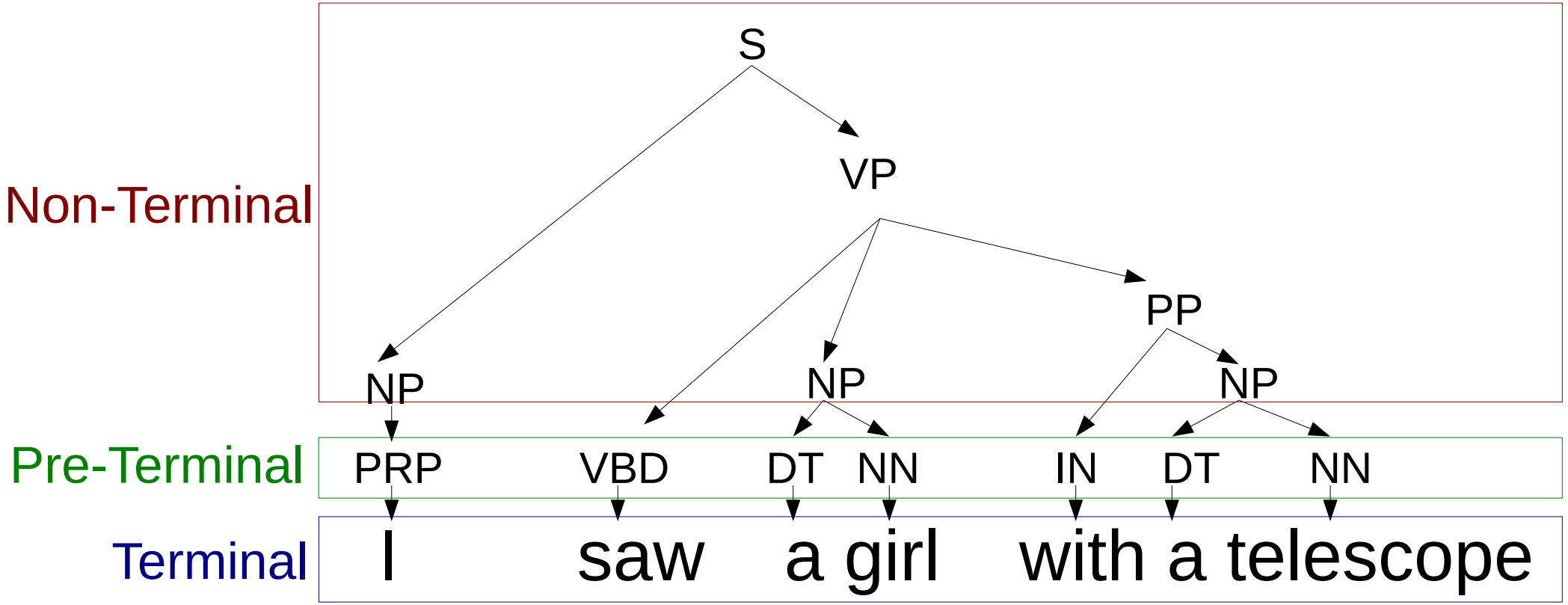
Different Structure, Different Interpretation



Different Structure, Different Interpretation

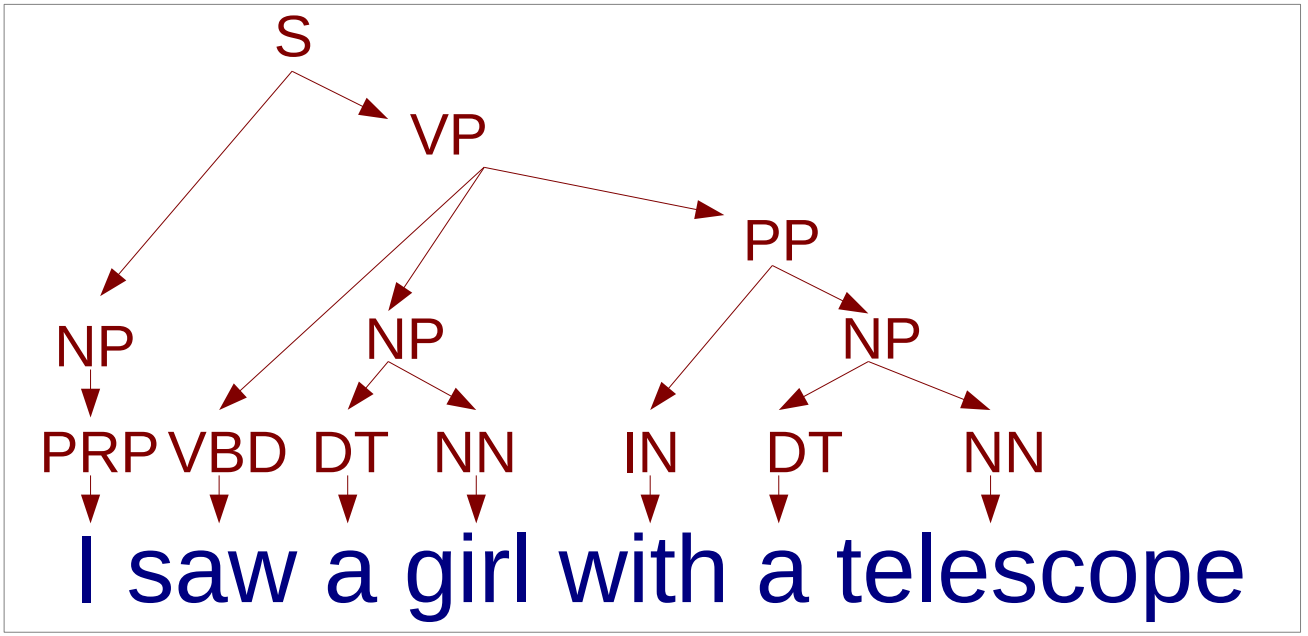


Non-Terminals, Pre-Terminals, Terminals



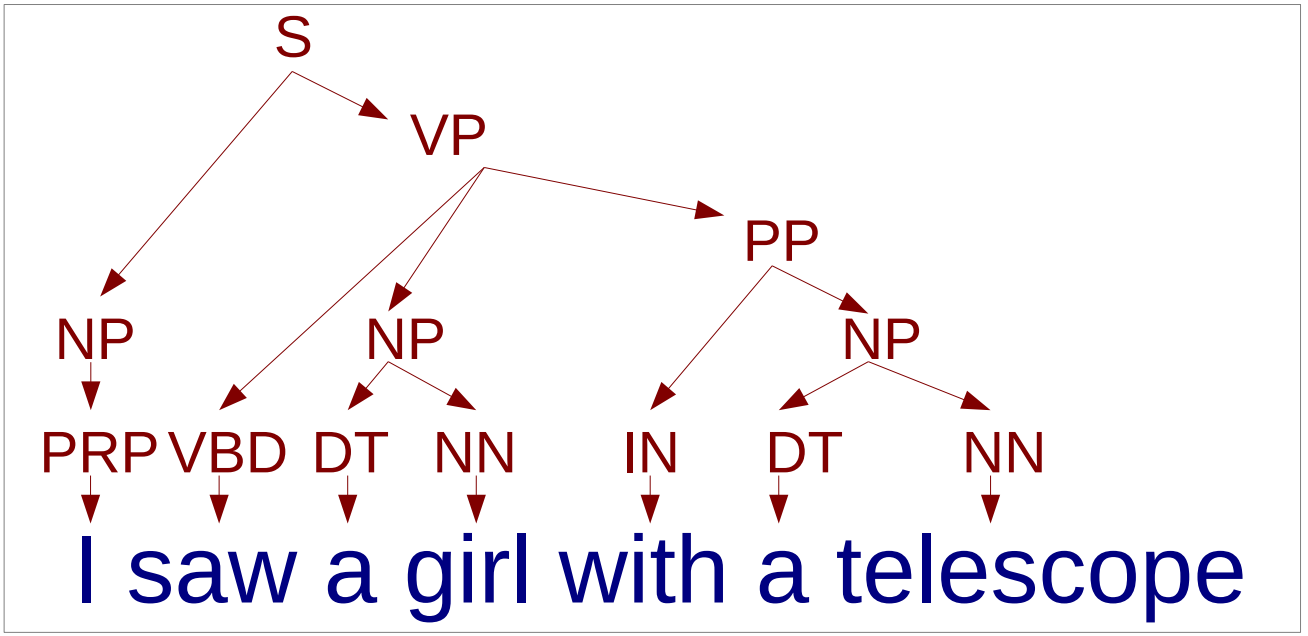
Parsing as a Prediction Problem

- Given a sentence X, predict its parse tree Y



Probabilistic Model for Parsing

- Given a sentence X, predict the most probable parse tree Y

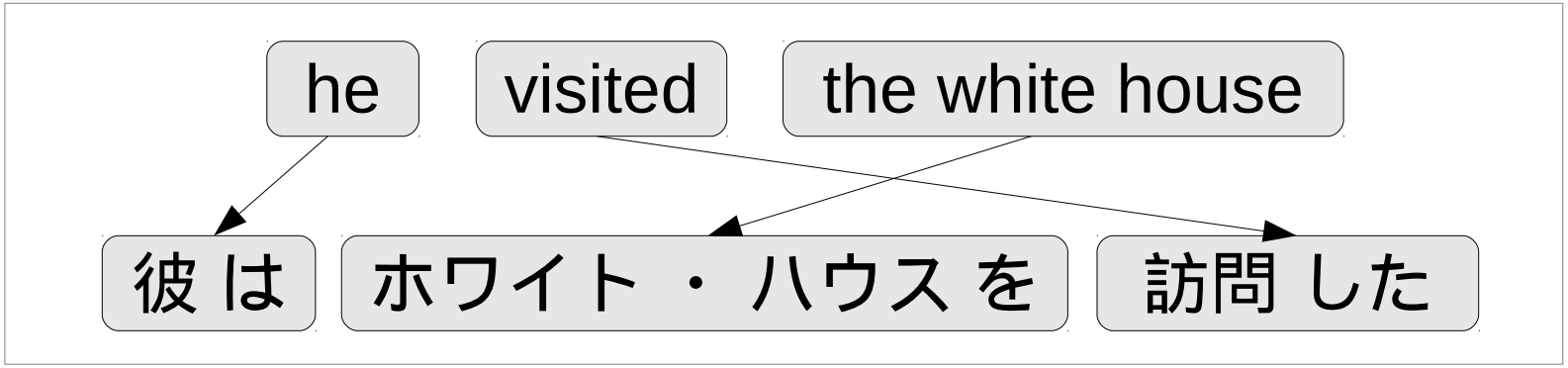


$$\underset{Y}{\operatorname{argmax}} P\left(Y \middle| X\right)$$

Hierarchical Phrase-based Machine Translation

Hierarchical PBMT (Hiero) [Chiang 07]

- Phrases are continuous



- Sometimes easier to use variables

example:

X_1 visited X_2 → X_1 は X_2 を 訪問 した

X_1 that was X_2 → X_2 で あった X_1

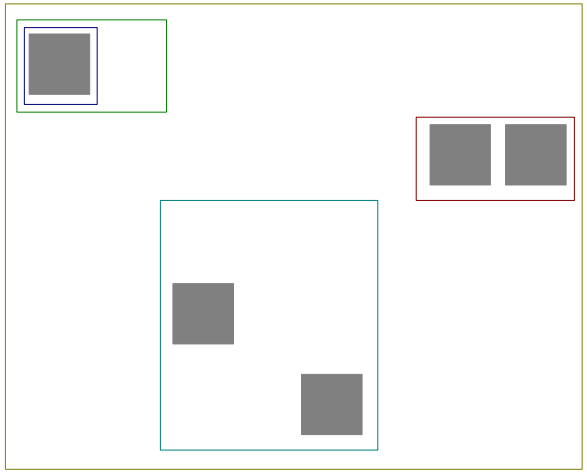
- Translation using this is “Hiero”

Changes from Phrase-Based

- Changes in Training: Extract non-contiguous phrases
- Changes in Translation: Algorithm similar to Viterbi, extended to HyperGraphs

Extracting Non-continuous Phrases

ホ
ワ　ハ
イ　ウ　訪し
彼はト・スを問た



1. First extract continuous

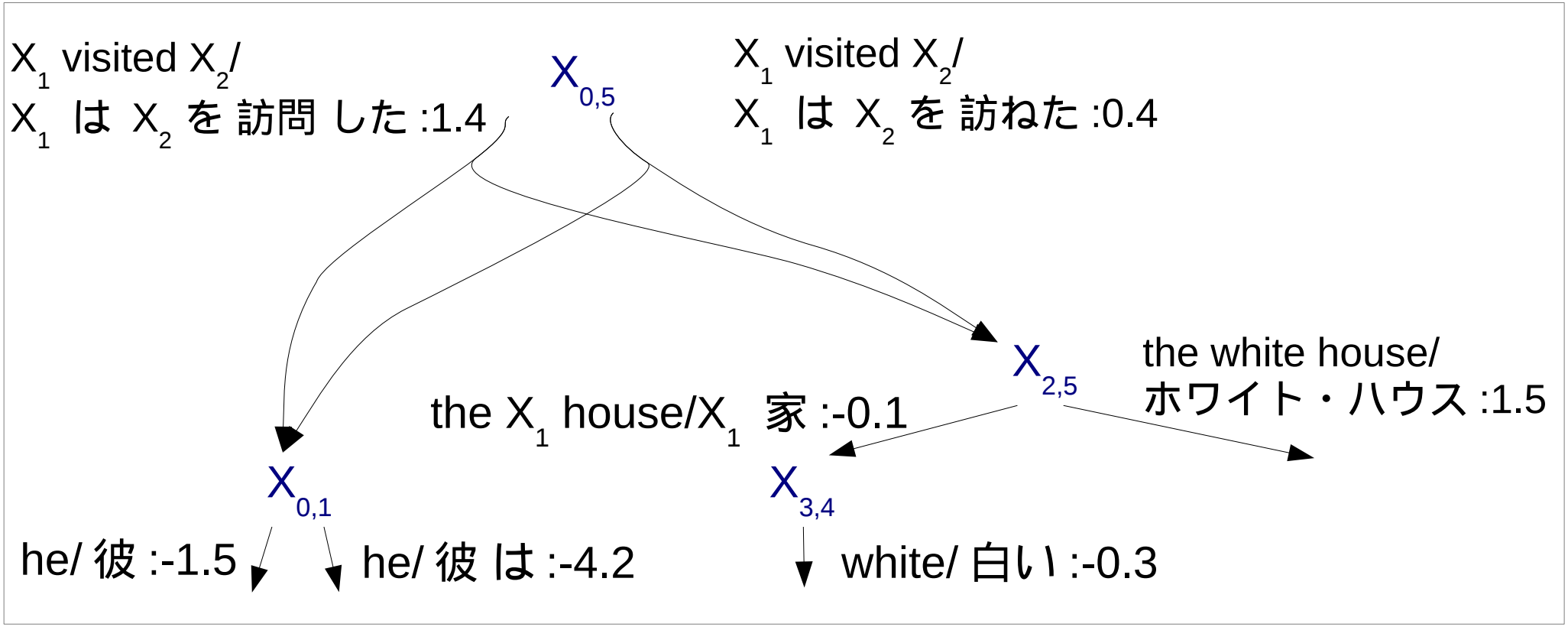
he → 彼
he → 彼 は
the white house → ホワイト・ハウス
visited → 訪問 した
he visited the white house
→ 彼 は ホワイト・ハウス を 訪問 した
…

2. Replace some of the alignments with variables

the X_1 house → X_1 ・ハウス
he X_1 the white house → 彼 は ホワイト・ハウス を X_1
he visited X_1 → 彼 は X_1 を 訪問 した
…

Hiero Translation

- Express the rules as a (hyper-) graph and choose which to use



Hiero Advatnages/Disadvantages

- + Better reordering accuracy
- - Slower translation
- - Larger models

Syntax-Based Translation

Syntax-based Translation

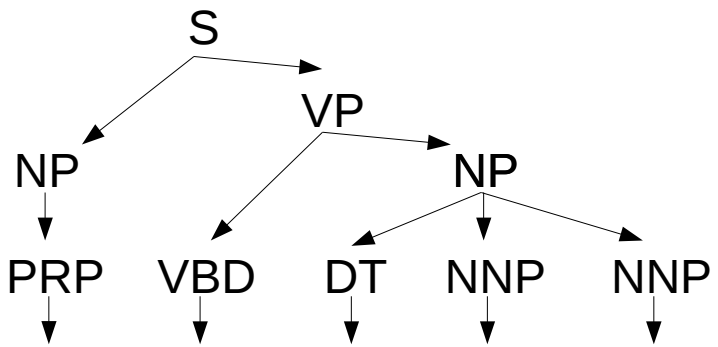
- Translation that actually uses syntactic information
- Parsing helps to identify phrases and reduce ambiguity
  →Can expect increases in accuracy
- Can both source and target syntax

Types of Syntax

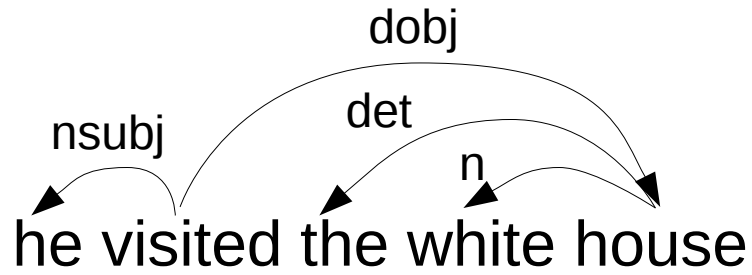
string

he visited the white house

tree



dependency

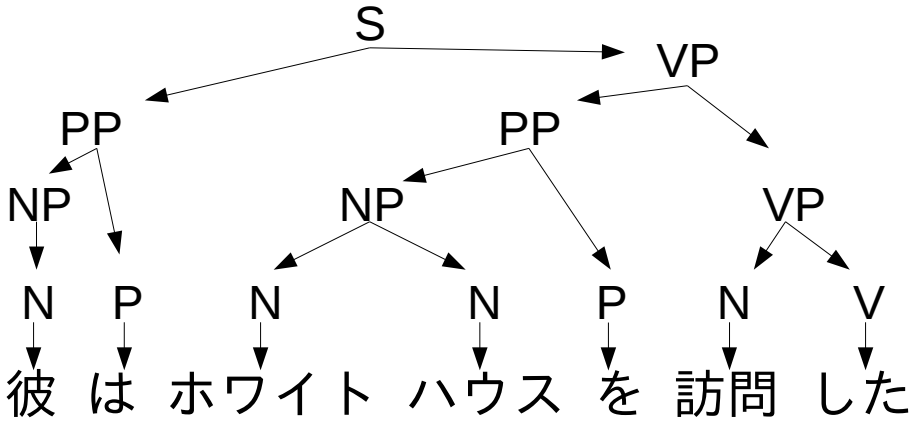


to

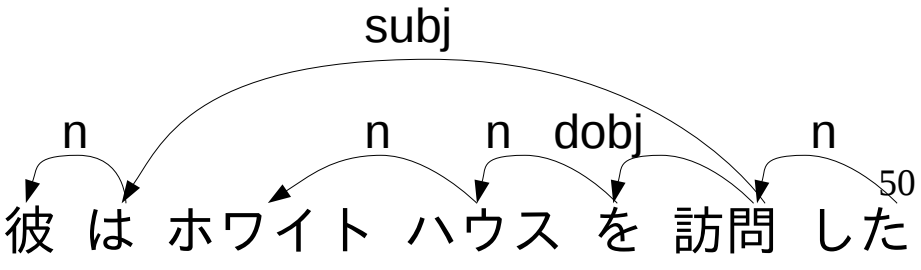
string

彼 は ホワイト ハウス を 訪問 した

tree



dependency



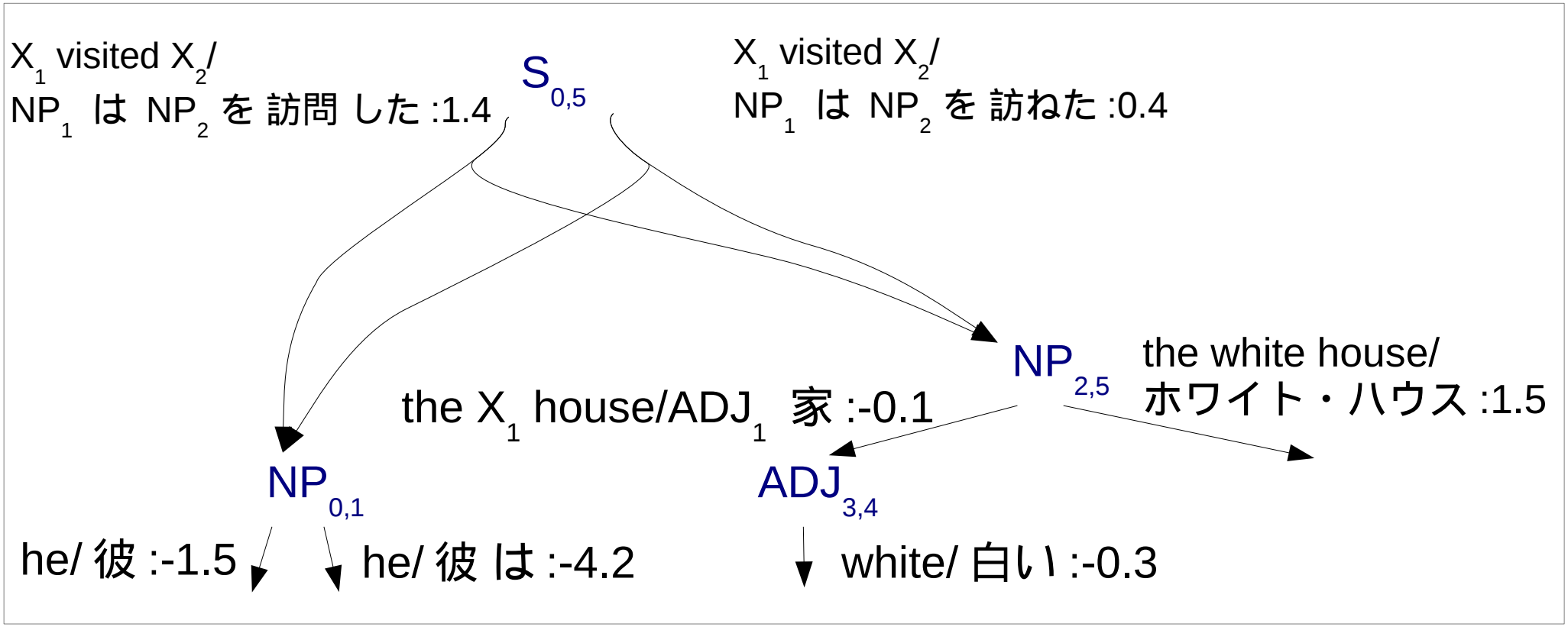
string-to-tree Translation [Galley+ 06]

- Use syntax only on the target side
- Basically the same as hierarchical translation
- Add labels to the target side

Source	Target	P	Score
he	彼	NP	-1.5
he	彼 は	NP	-4.2
X_1 visited X_2	NP_1 は NP_2 を 訪問 した	S	1.4
X_1 visited X_2	NP_1 は NP_2 を 訪ねた	S	0.4
the white house	ホワイト ・ ハウス	NP	1.5
the X_1 house	ADJ_1 家	NP	-0.1
white	白い	ADJ	-0.3

String-to-tree translation

- Consider the target labels during translation



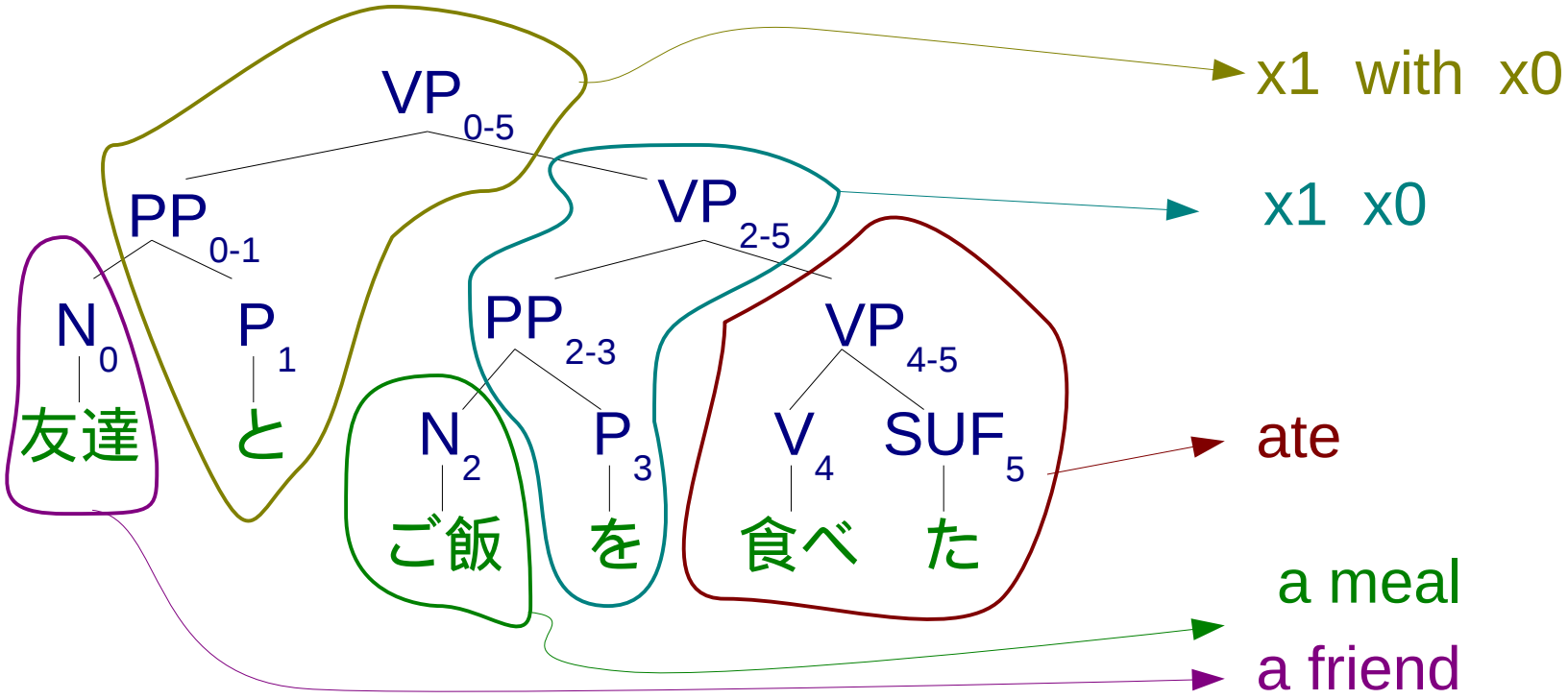
- Cannot use rules that don't match the syntax (cannot insert an NP into ADJ)

tree-to-string Translation

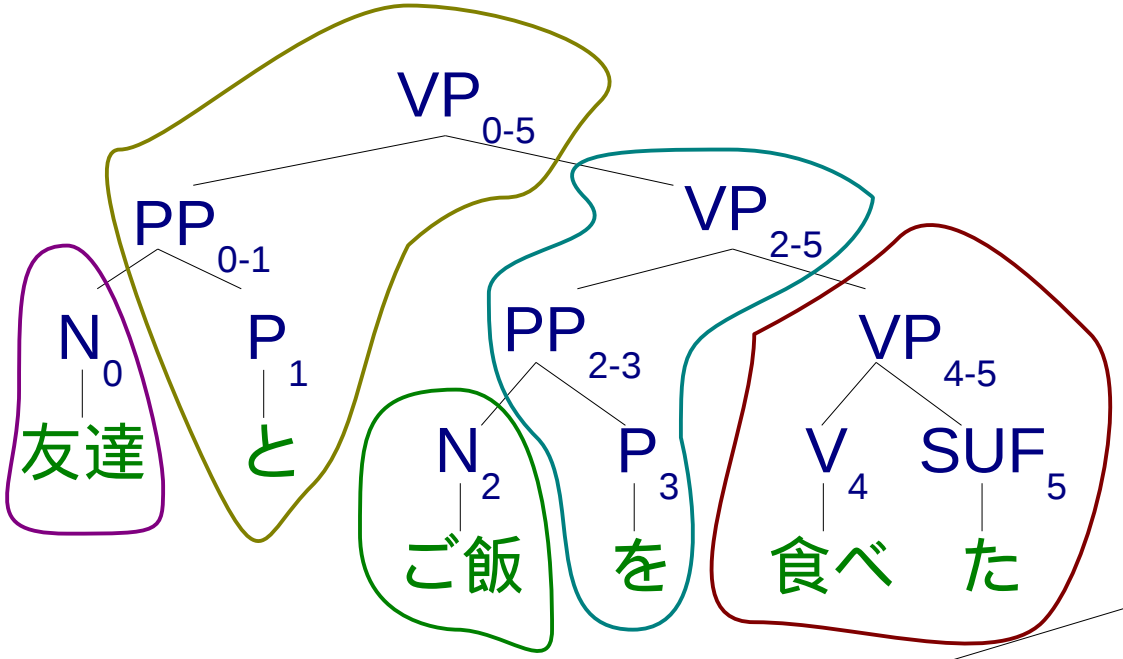
- Use syntactic information on the source side
- Mainly perform parsing before starting translation
  - + Fast
  - + Has less problem with long distance reorering
  - - Heavily affected by parsing mistakes

tree-to-string Translation [Liu+ 06]

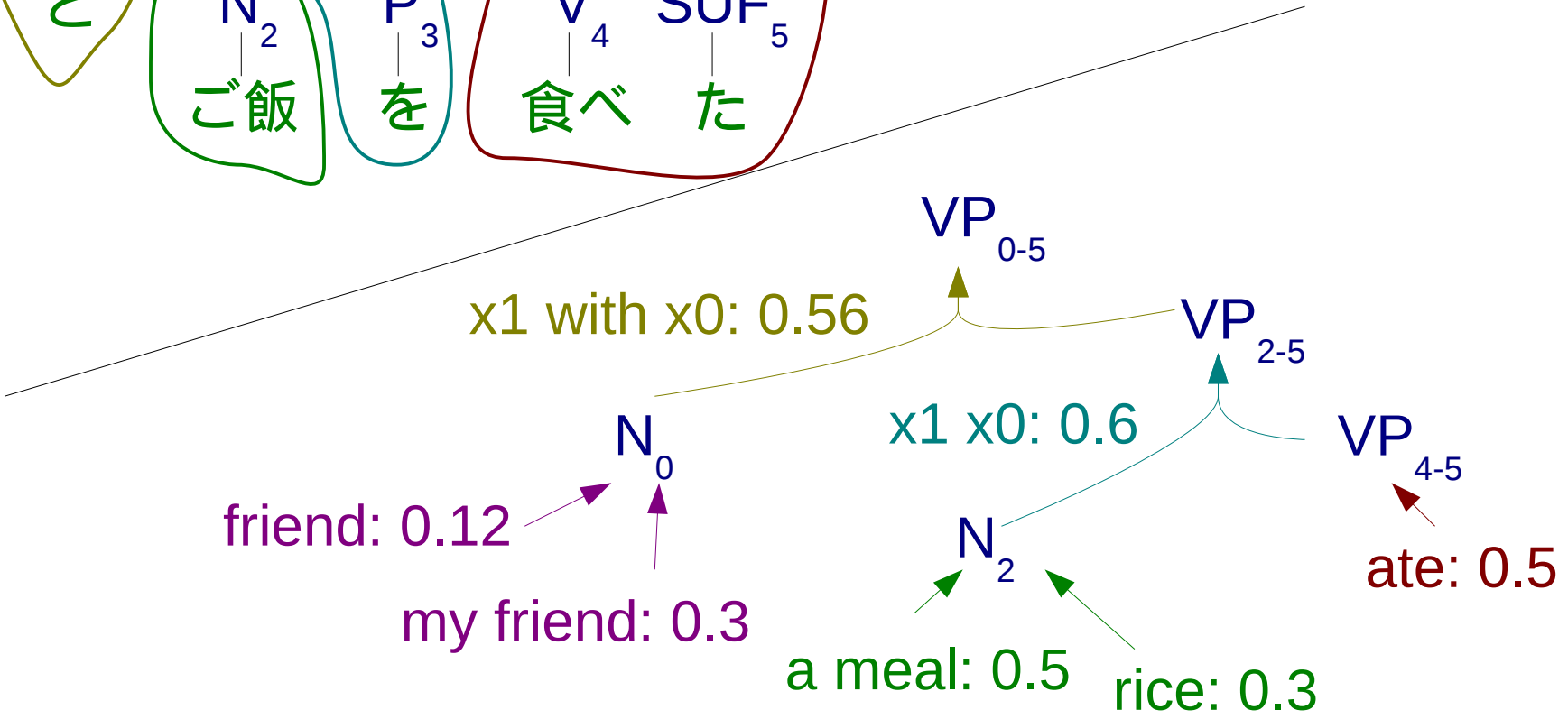
- Match parts of the parse tree, and translate them



tree-to-string Translation [Liu+ 06]



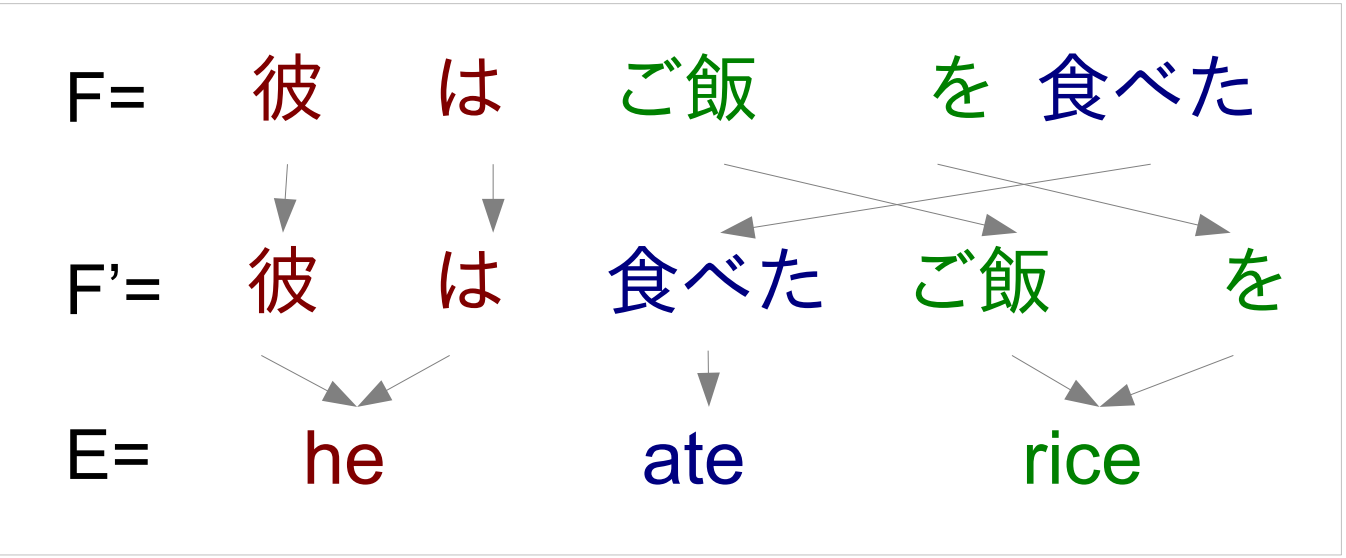
- Create a graph representing rules
- Search is the same as Hiero



Pre-ordering

Pre-ordering [Xia+ 04]

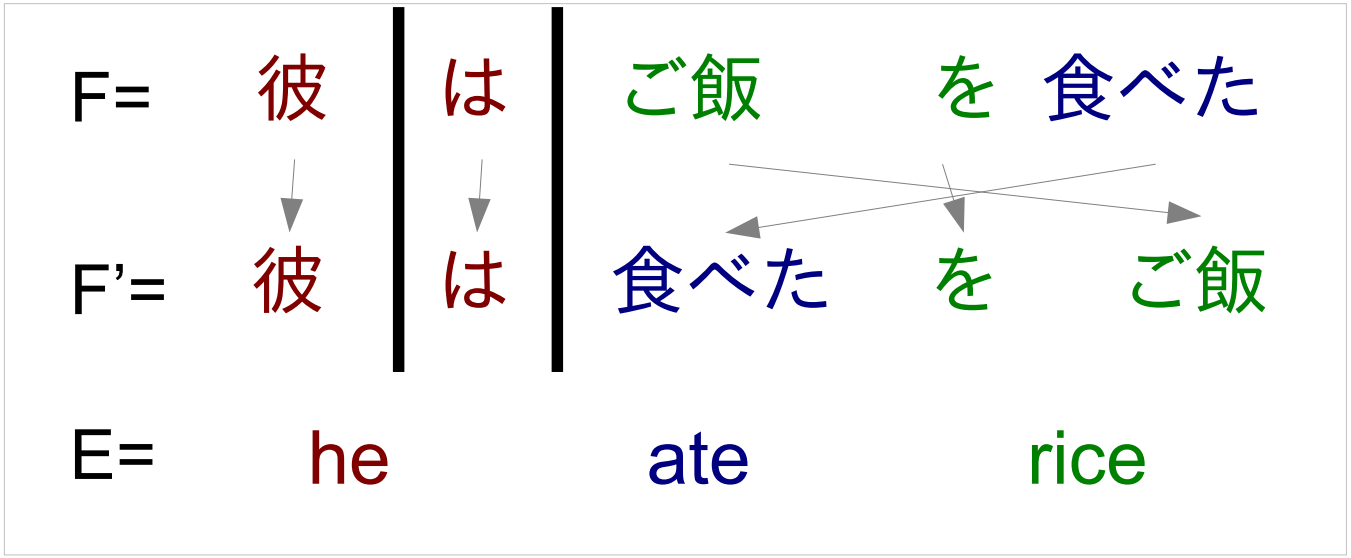
- Phrase-based translation is strong, but not very good at long-distance reordering
- Pre-ordering first reorders the source into the target order



- Note: pre-order before training as well

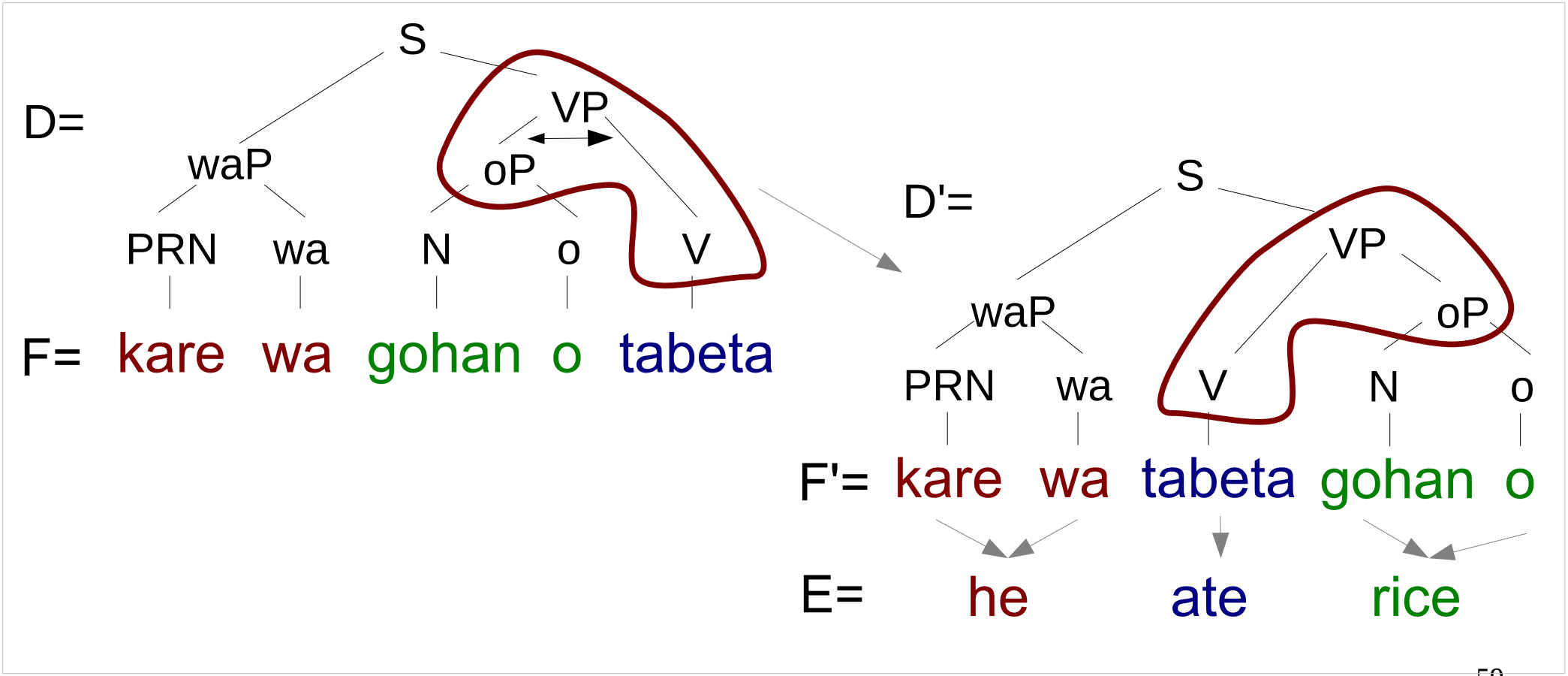
Heuristics-based Preordering [Katz-Brown+ 08]

- For example, in Japanese English, use the following simple rule
  - Keep “ は” and punctuation in the same place
  - Reverse the order of everything else



Syntax-based Pre-ordering

- Create a parse of the source, and re-order the parse tree



Head Finalization [Isozaki+ 10]

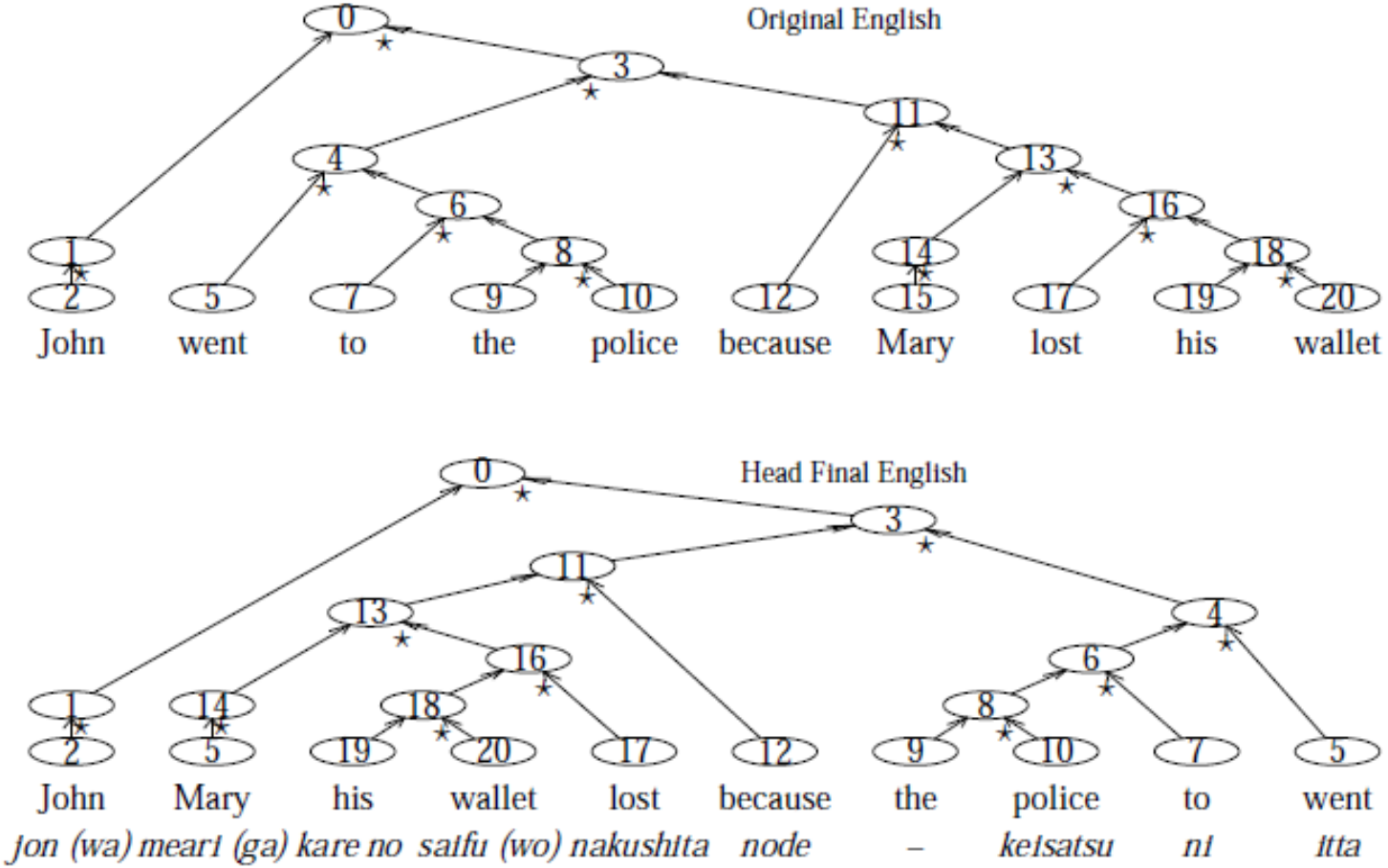


Figure 3: Head-Finalizing a complex sentence.

Assignment

Assignment

- (Only one assignment this week)
- You are given a baseline machine translation system
  - LM/Alignment: Baseline from exercises 1, 2
  - TM: Phrases of up to length 4
  - SM: Uniform distribution
  - RM: Distortion penalty
  - Reordering Limit: 6
- Try to improve its accuracy by changing one of the features listed above, or anything else

Files to Look At

- Alignment: Substitute your files in the pipeline
- Phrase Extraction: Modify phrase-extract.py
- LM: Modify the part of “decoder.py” that calculates the LM probability
- Tuning or Decoding Settings: You can adjust the parameters at the top of decoder.py
- Preordering: Pre-order the source side of the training and testing files before running everything

Assignment Details

- Download the exercise from the web
- You can find a list of commands to run in run-translate.sh
- Send any files you changed, BLEU score before/after, and a short description of the change
  - Due date: February 12th, 23:59
  - Address: neubig@is.naist.jp