ANTONIS ANASTASOPOULOS
CS499 INTRODUCTION TO NLP

NEURAL MODELS FOR DEPENDENCY PARSING

https://cs.gmu.edu/~antonis/course/cs499-spring21/

With adapted slides by Graham Neubig

LOGISTICS

Today: in-class exercise

Friday: Project Baseline Due

https://cs.gmu.edu/~antonis/course/cs499-spring21/project/

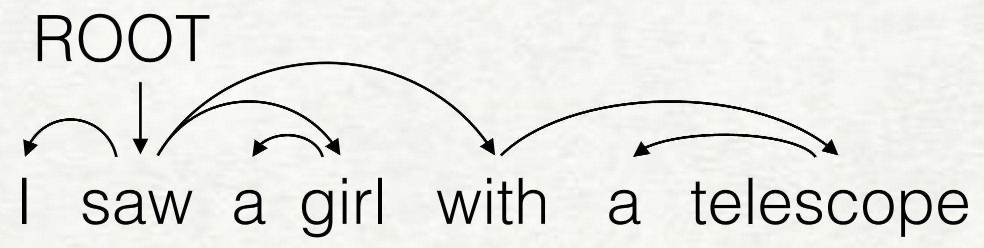
Coming up: Project Presentations

STRUCTURE OF THIS LECTURE

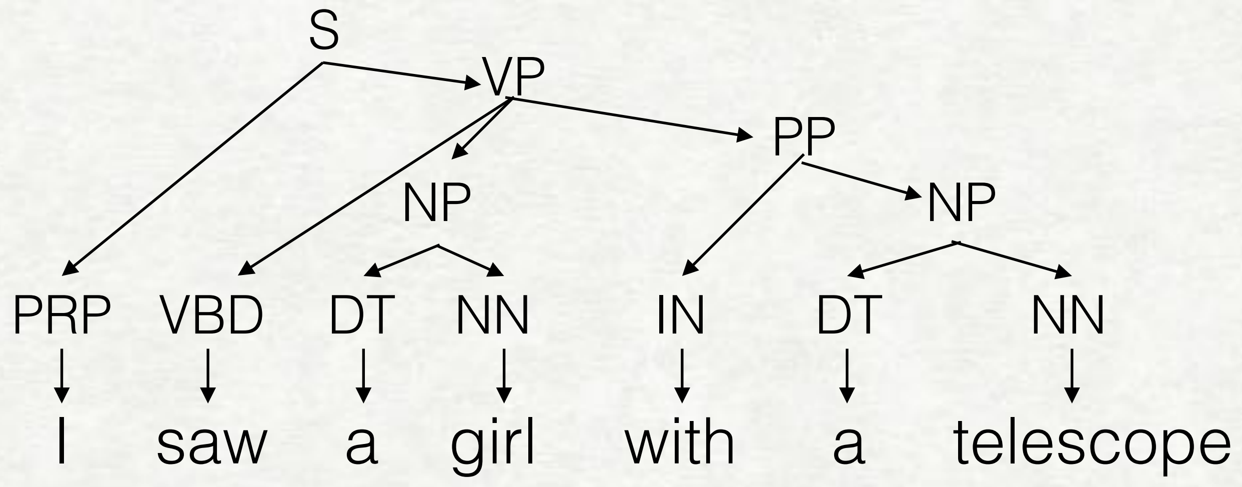
1. Dependency Parsing
2. Shift-Reduce Methods
3. Graph-Based Methods
4. Multilingual Parsing

TWO TYPES OF LINGUISTIC STRUCTURE

Dependency: focus on relations between words

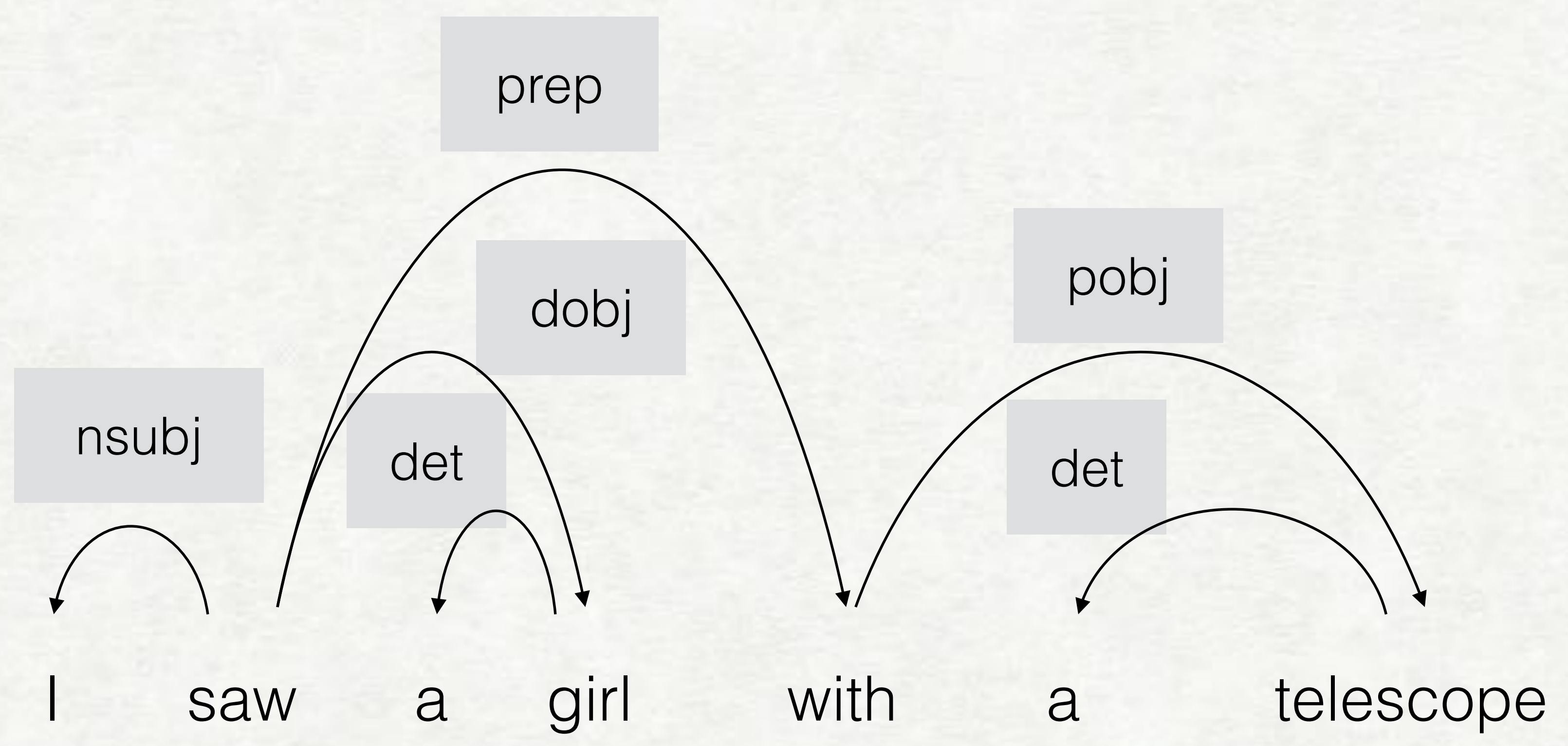


Phrase Structure: focus on the structure of the sentence



WHY DEPENDENCIES?

1. Demonstrate the relationships between words in a straightforward way



2. Particularly good for multilinguality, because phrase structure can be hard to define in languages with free word order

UNIVERSAL DEPENDENCIES TREEBANK

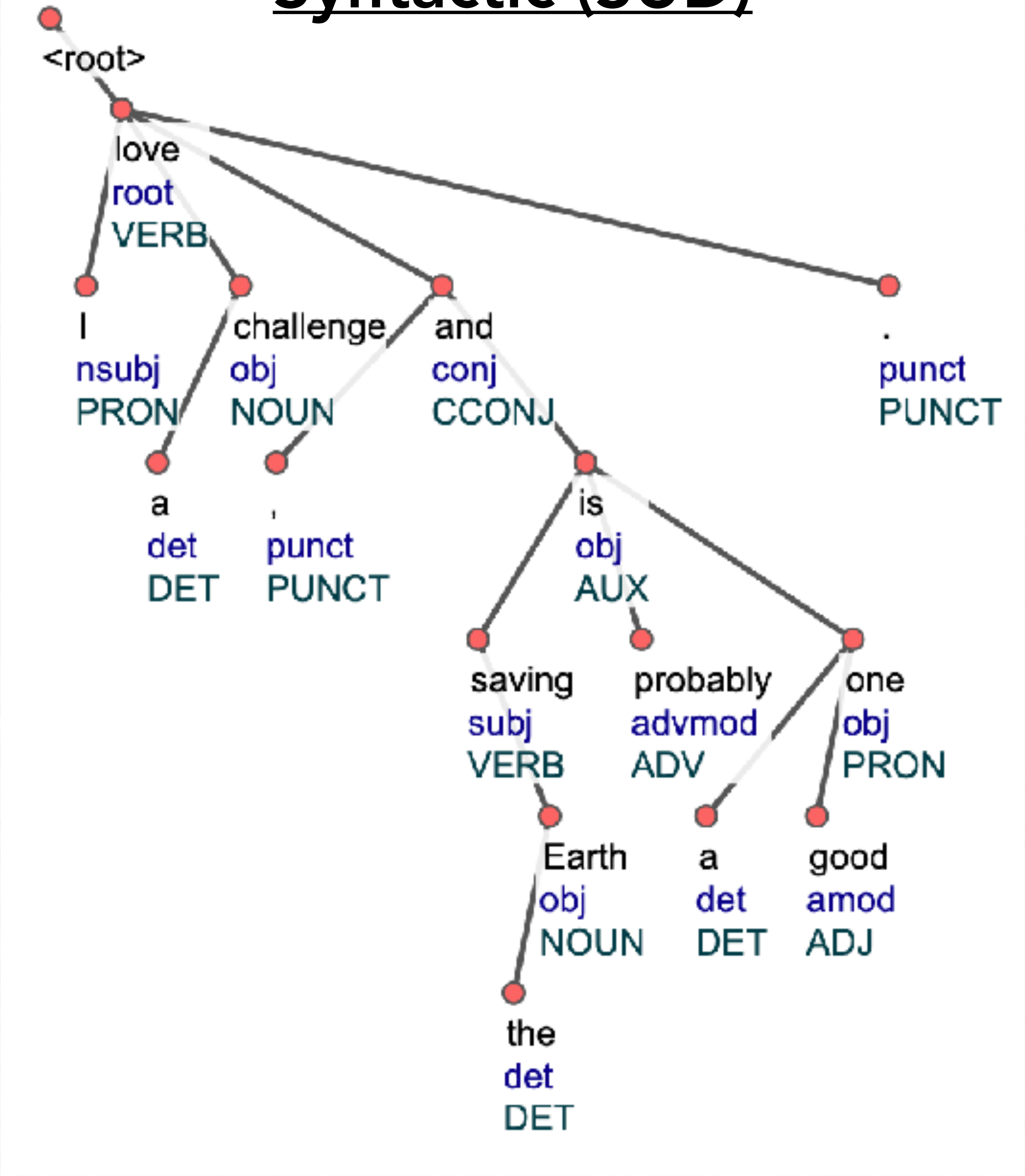
Standard format for parse trees in many languages

Abaza	1	3K	Northwest Caucasian
Afrikaans	1	49K	IE, Germanic
Akkadian	1	1K	Afro-Asiatic, Semitic
Albanian	1	<1K	IE, Albanian
Amharic	1	10K	Afro-Asiatic, Semitic
Ancient Greek	2	416K	IE, Greek
Arabic	3	1,042K	Afro-Asiatic, Semitic
Armenian	1	52K	IE, Armenian
Assyrian	1	<1K	Afro-Asiatic, Semitic
Bambara	1	13K	Mande
Basque	1	121K	Basque
Belarusian	1	13K	IE, Slavic
Bhojpuri	2	6K	IE, Indic
Breton	1	10K	IE, Celtic
Bulgarian	1	156K	IE, Slavic
Buryat	1	10K	Mongolic
Cantonese	1	13K	Sino-Tibetan

https://universaldependencies.org/

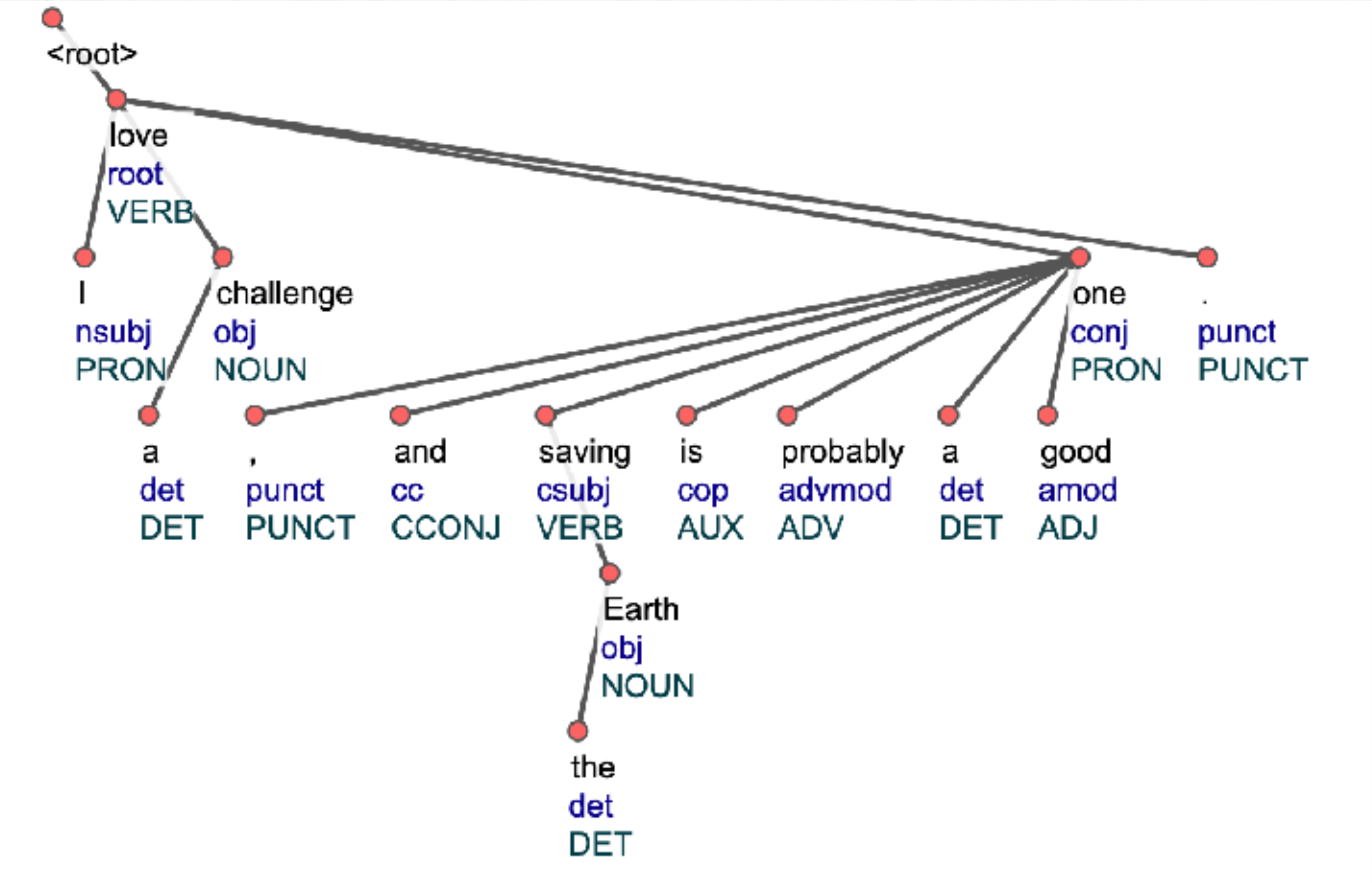
SEMANTIC AND SYNTACTIC DEPENDENCIES

<u>Syntactic (SUD)</u>



Deeper, reflect phrase structure, more function word heads
https://surfacesyntacticud.github.io/

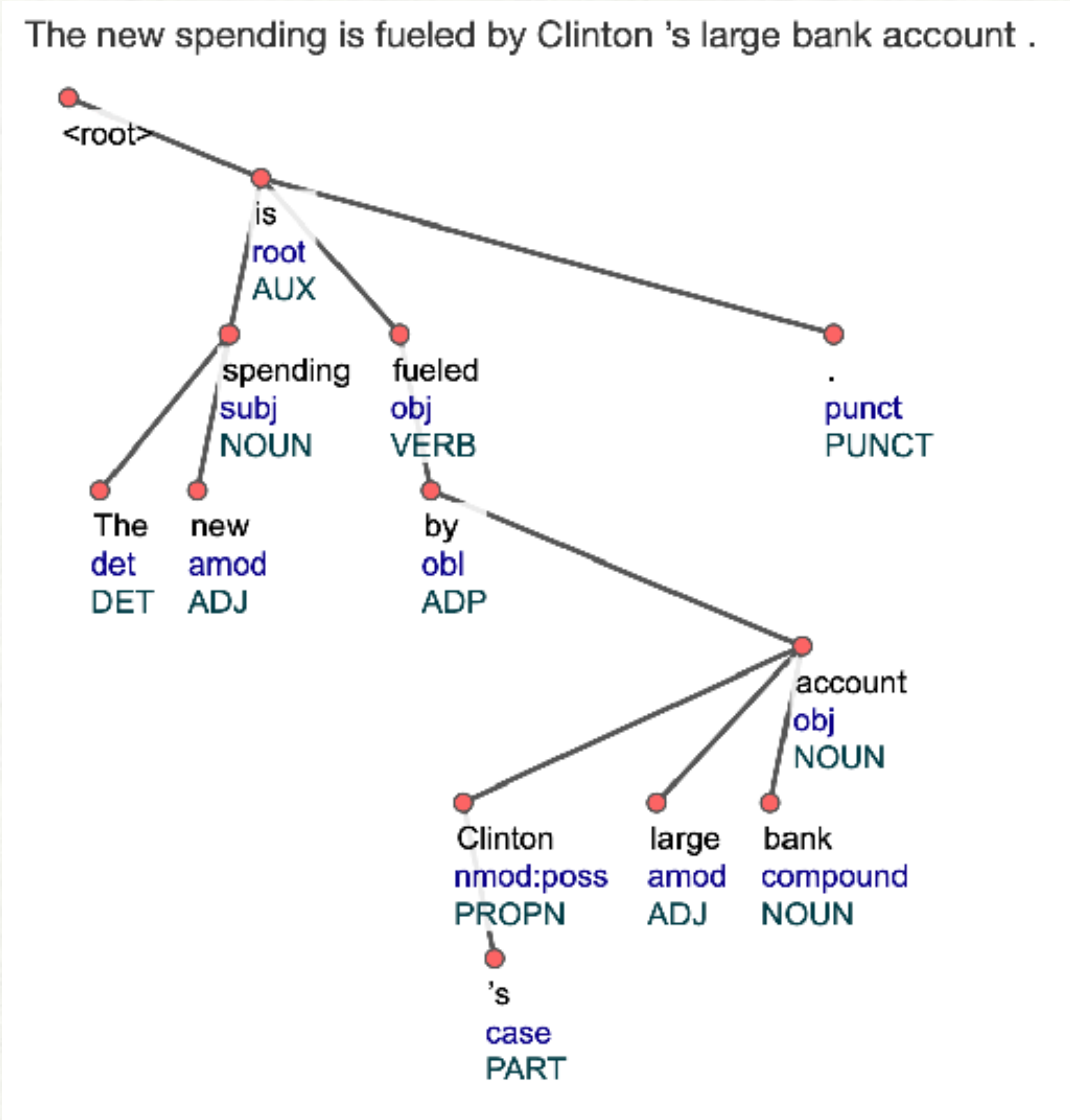
<u>Semantic (UD)</u>



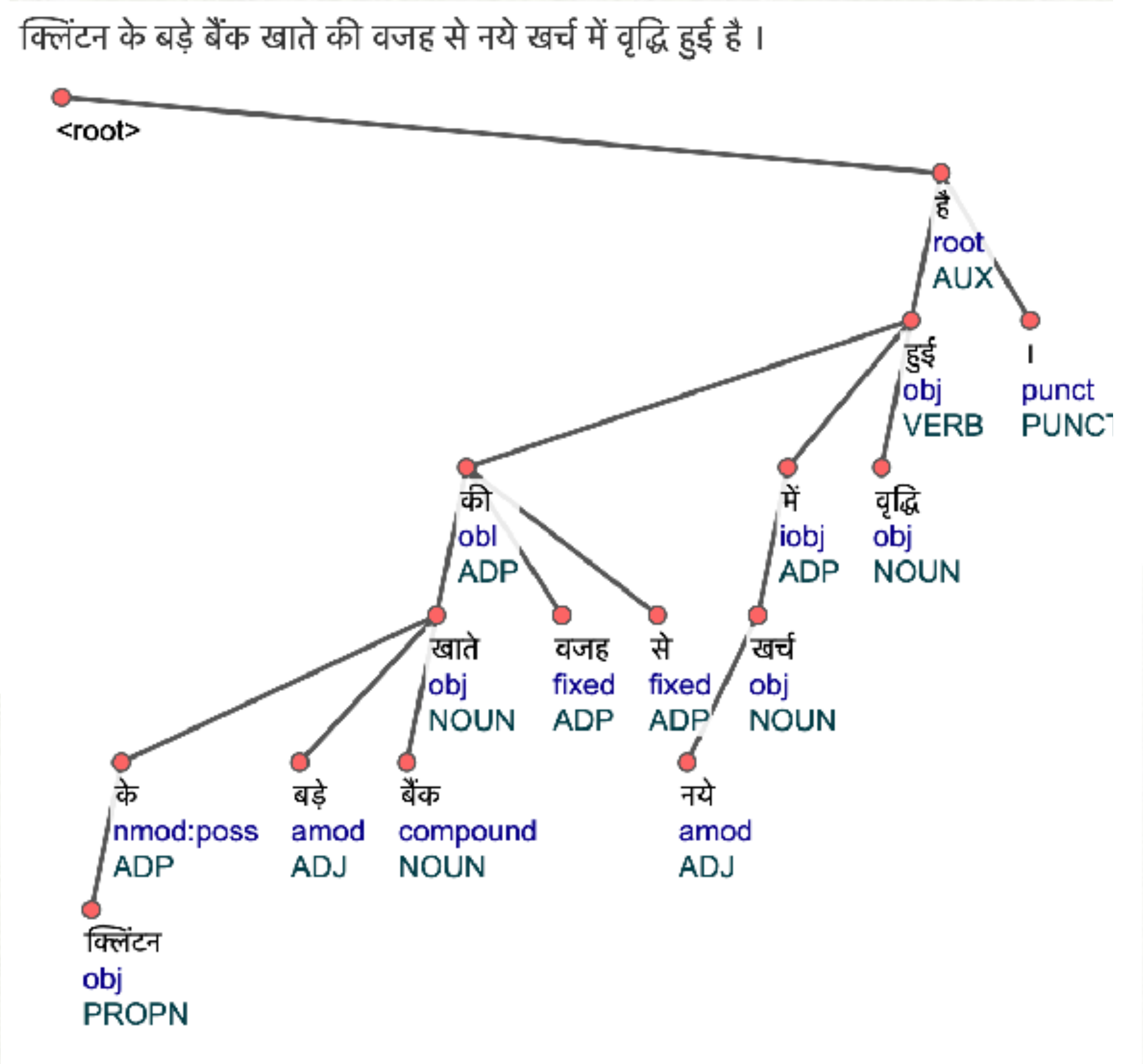
Flatter, semantically related words closer, more content word heads

CROSS-LINGUAL DIFFERENCES IN STRUCTURE

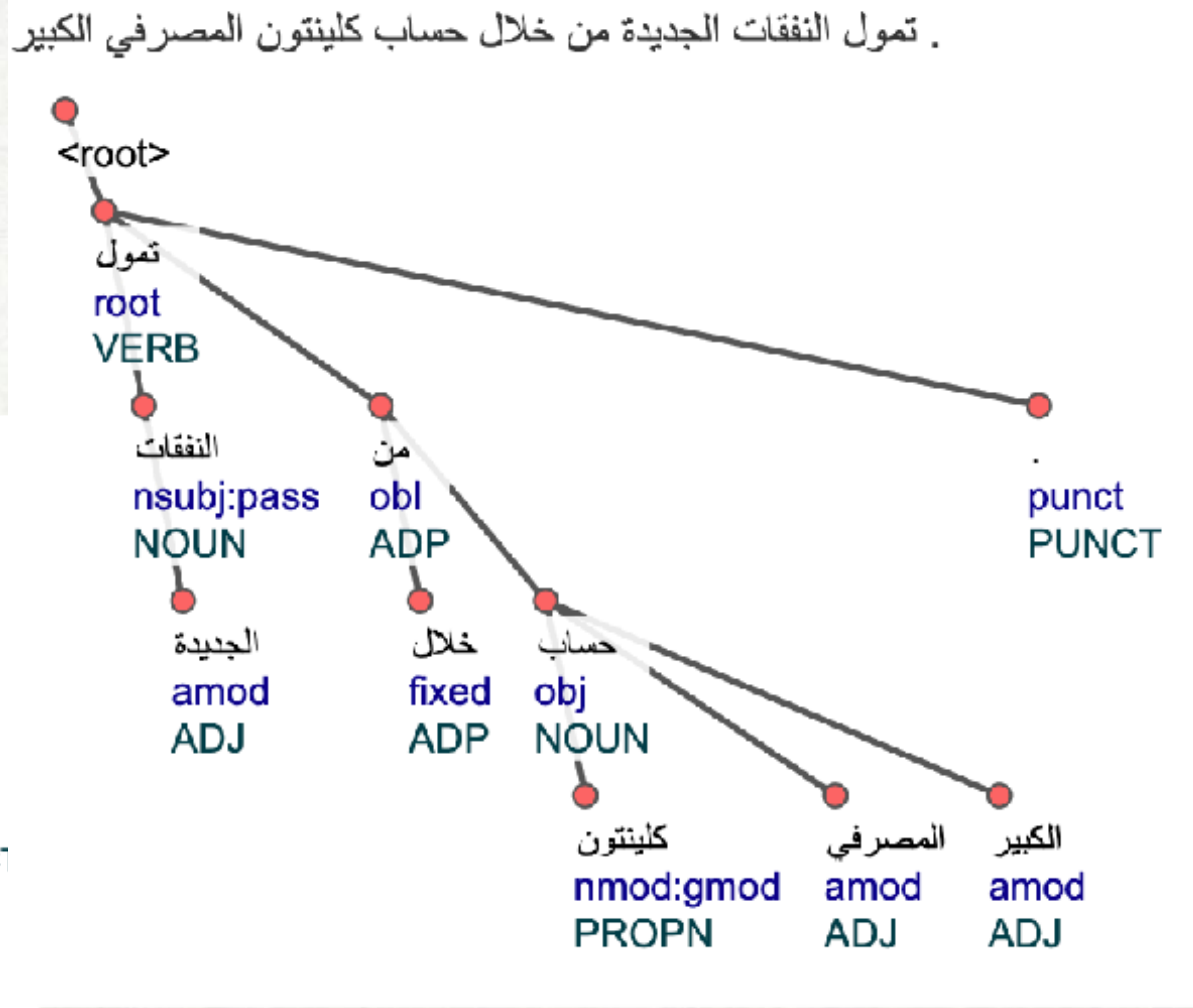
English: SVO



Hindi: Verb Final



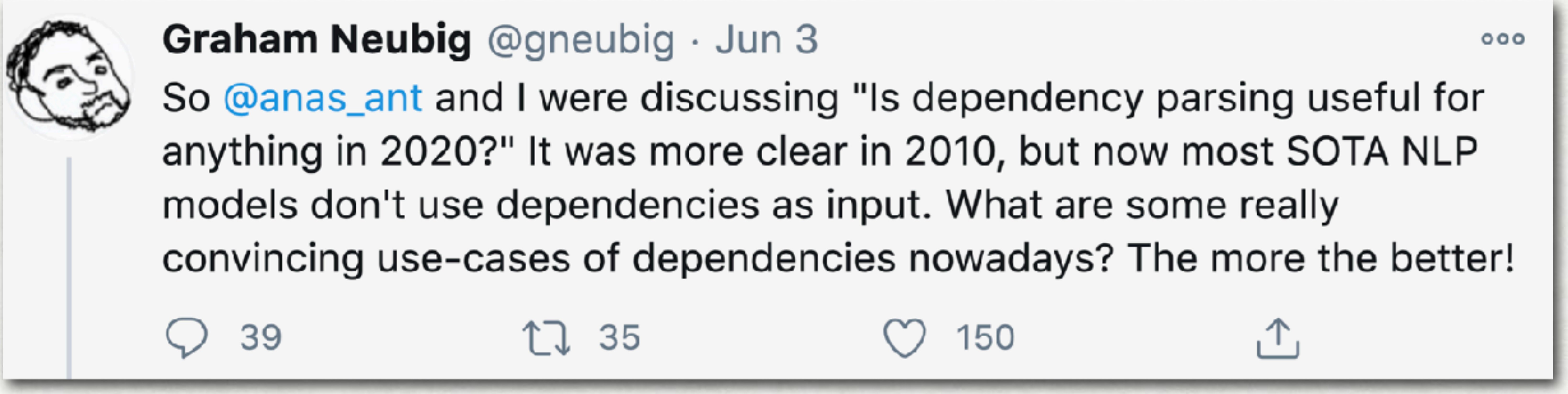
Arabic: Verb Initial



USE CASES OF DEPENDENCIES?

Previously, used for feature engineering in systems (and still useful in some cases)

Now: more useful for human-facing applications


Graham Neubig @gneubig · Jun 3

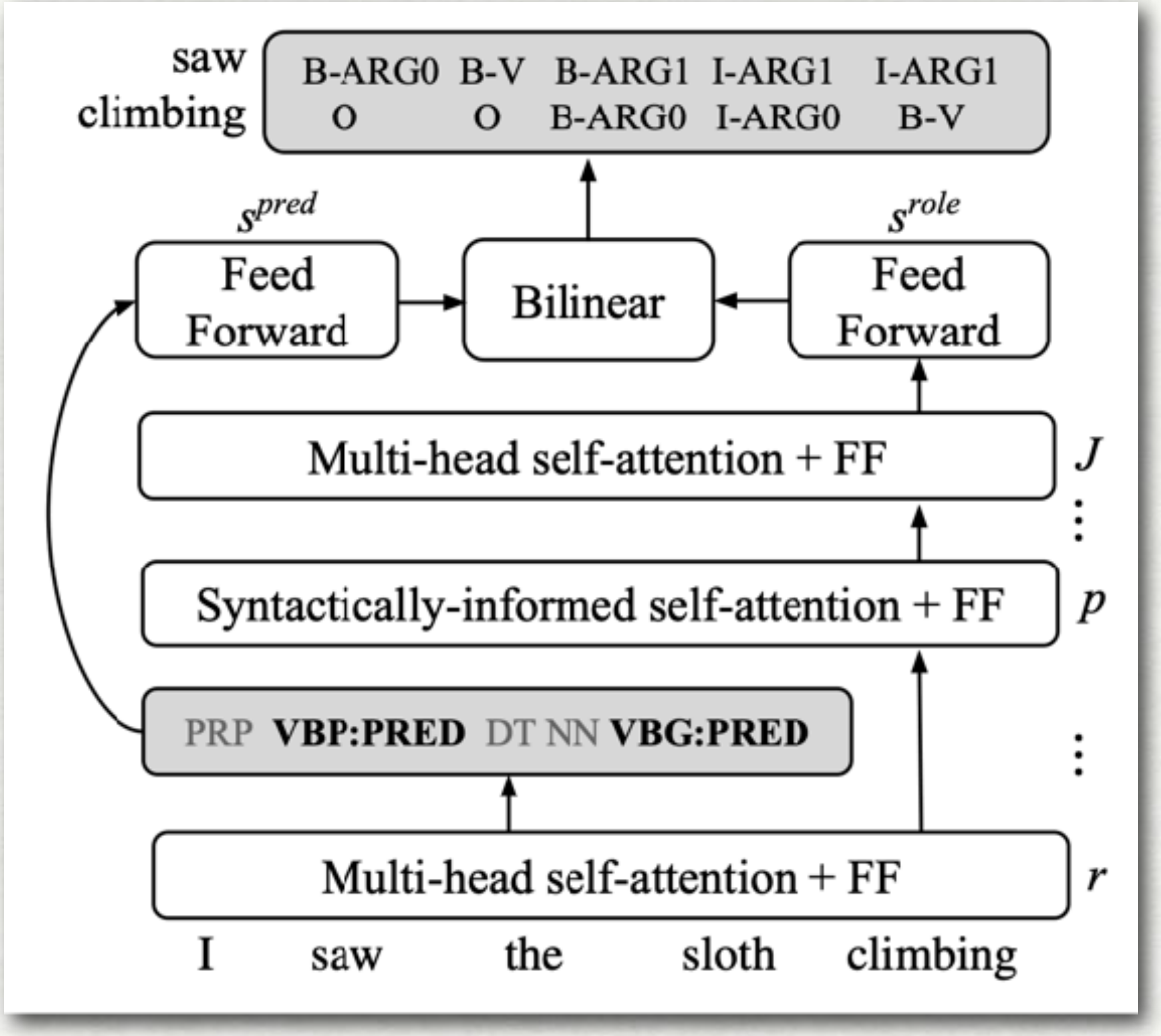
So @anas_ant and I were discussing "Is dependency parsing useful for anything in 2020?" It was more clear in 2010, but now most SOTA NLP models don't use dependencies as input. What are some really convincing use-cases of dependencies nowadays? The more the better!

39 35 150

https://twitter.com/gneubig/status/1268238606101032962?lang=en

EXAMPLE 1: ADDING INDUCTIVE BIAS TO NEURAL MODELS

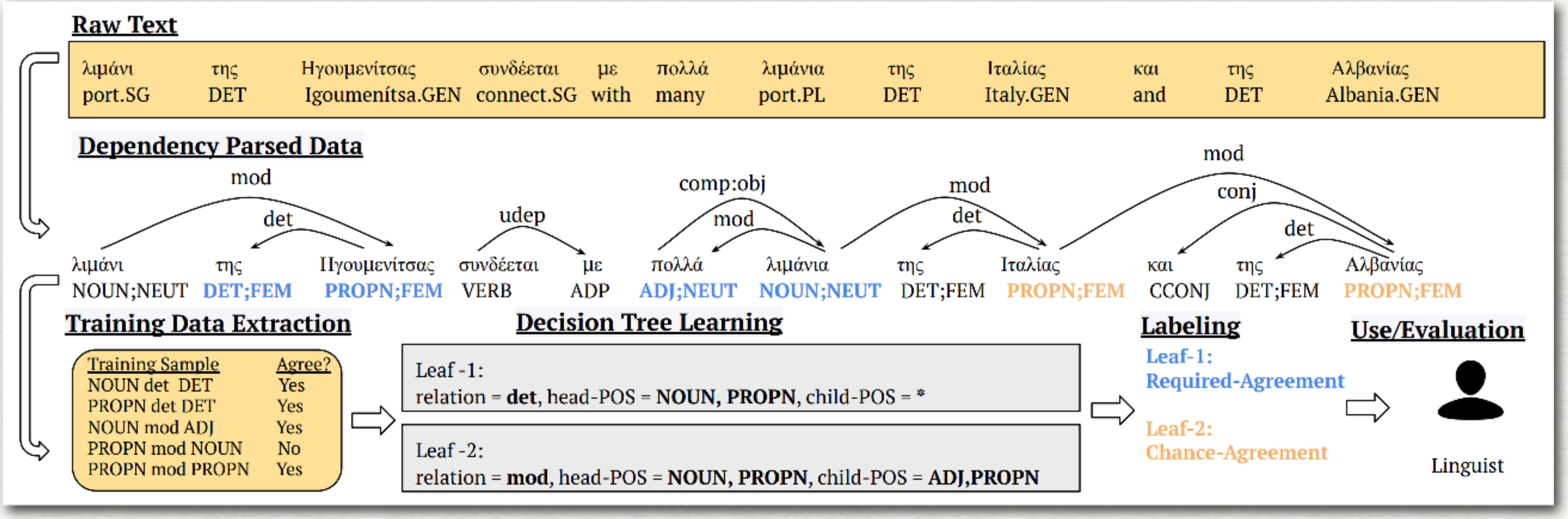
Bias self attention to follow syntax



Strubell, Emma, et al. "Linguistically-informed self-attention for semantic role labeling." arXiv preprint arXiv:1804.08199 (2018).

EXAMPLE 2: UNDERSTANDING LANGUAGE STRUCTURE

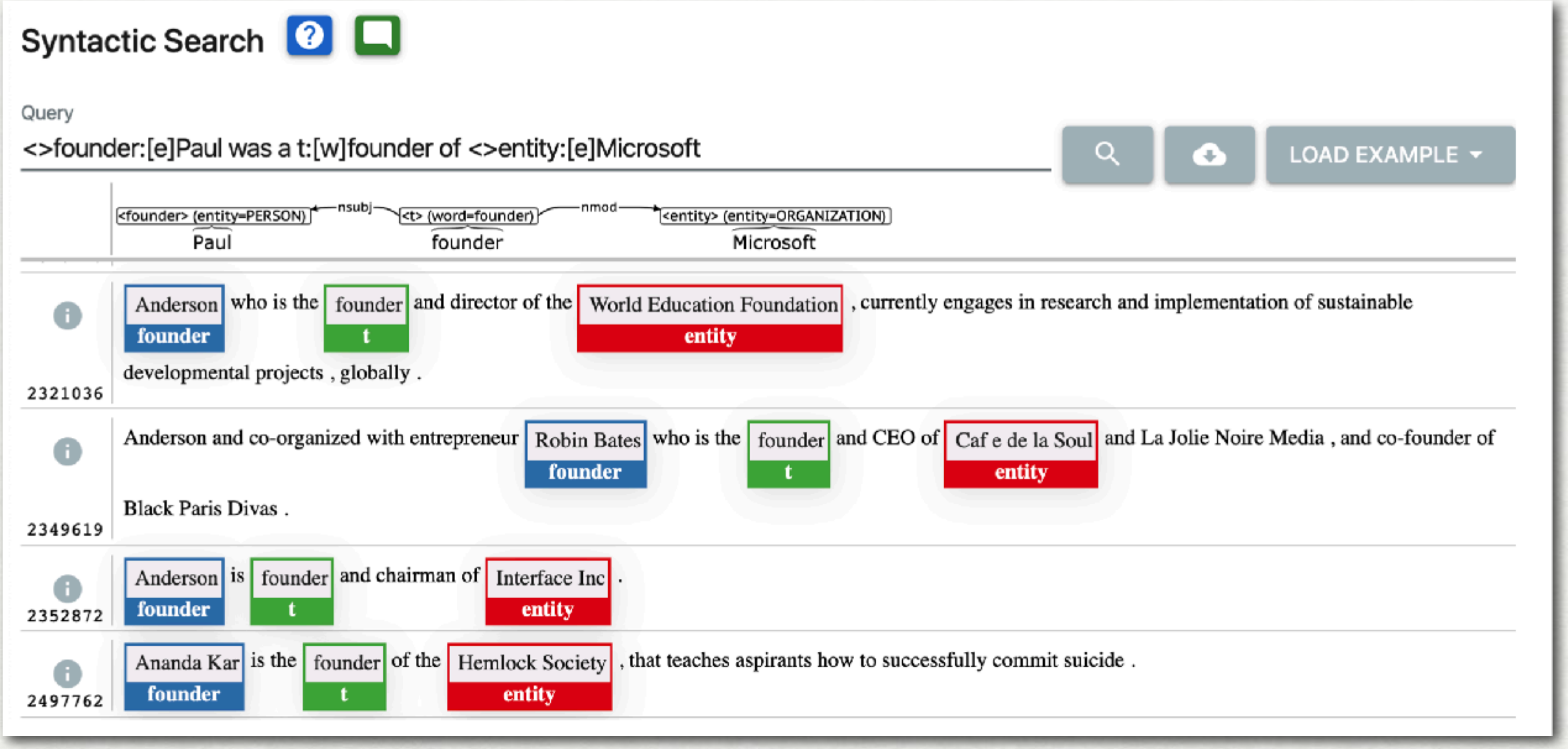
Example of extracting morphological agreement rules using dependency relations



Chaudhary, Aditi, et al. "Automatic Extraction of Rules Governing Morphological Agreement." EMNLP 2020.

EXAMPLE 3: SEARCHING OVER PARSED CORPORA

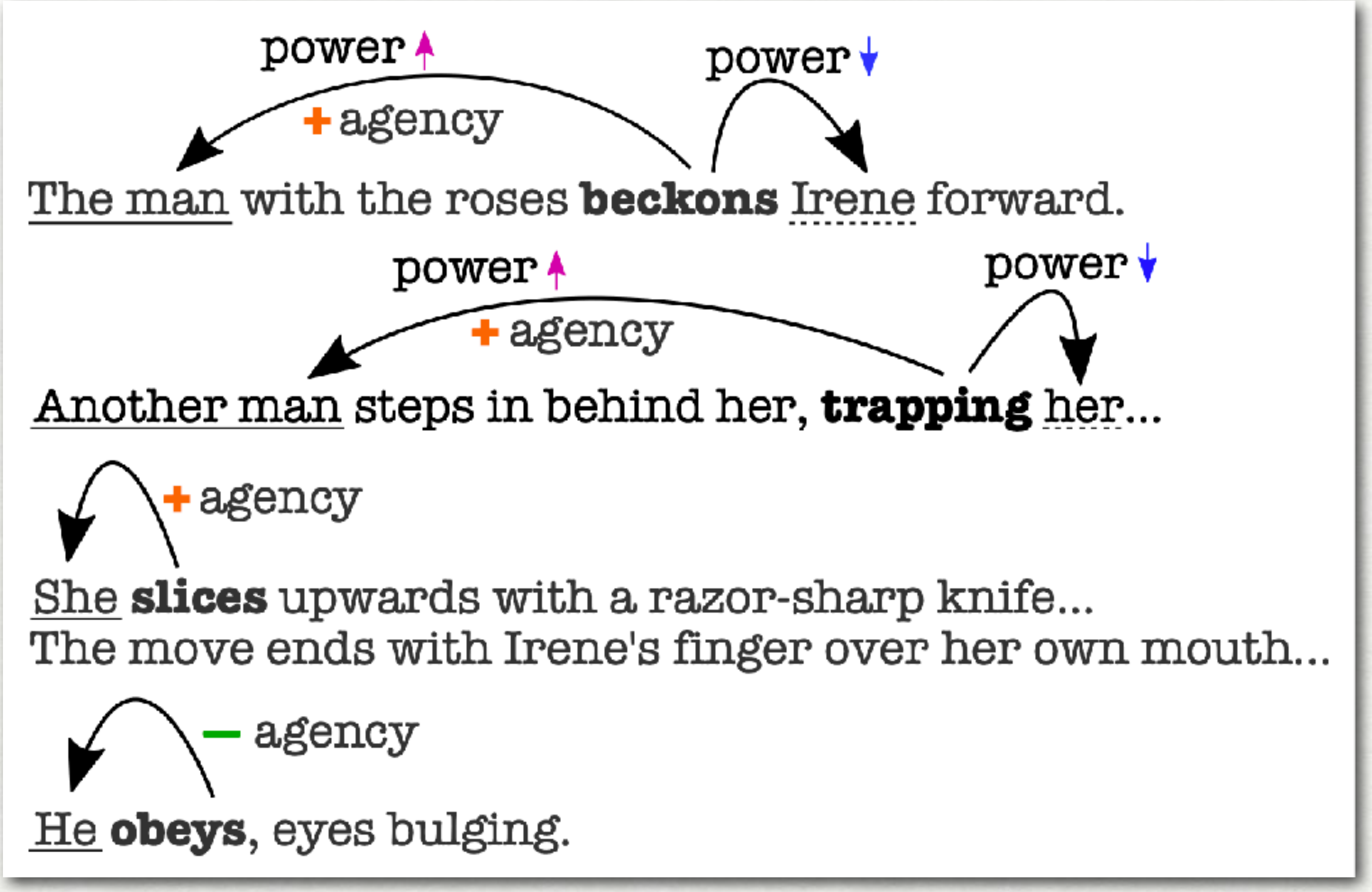
Search using "syntactic regex"



Shlain, Micah, et al. "Syntactic Search by Example." arXiv preprint arXiv:2006.03010 (2020).

EXAMPLE 4: ANALYSIS OF OTHER LINGUISTIC PHENOMENA

Examining power and agency in film scripts



Sap, Maarten, et al. "Connotation frames of power and agency in modern films." EMNLP 2017.

EXERCISE

DEPENDENCY PARSING

PARSING

Predicting linguistic structure from input sentence

Transition-based models

- step through actions one-by-one until we have output
- like history-based model for POS tagging

Graph-based models

- calculate probability of each edge/constituent, and perform some sort of dynamic programming
- like linear CRF model for POS

SHIFT-REDUCE PARSING

ARC STANDARD SHIFT-REDUCE PARSING (YAMADA & MATSUMOTO 2003, NIVRE 2003)

Process words one-by-one left-to-right

Two data structures

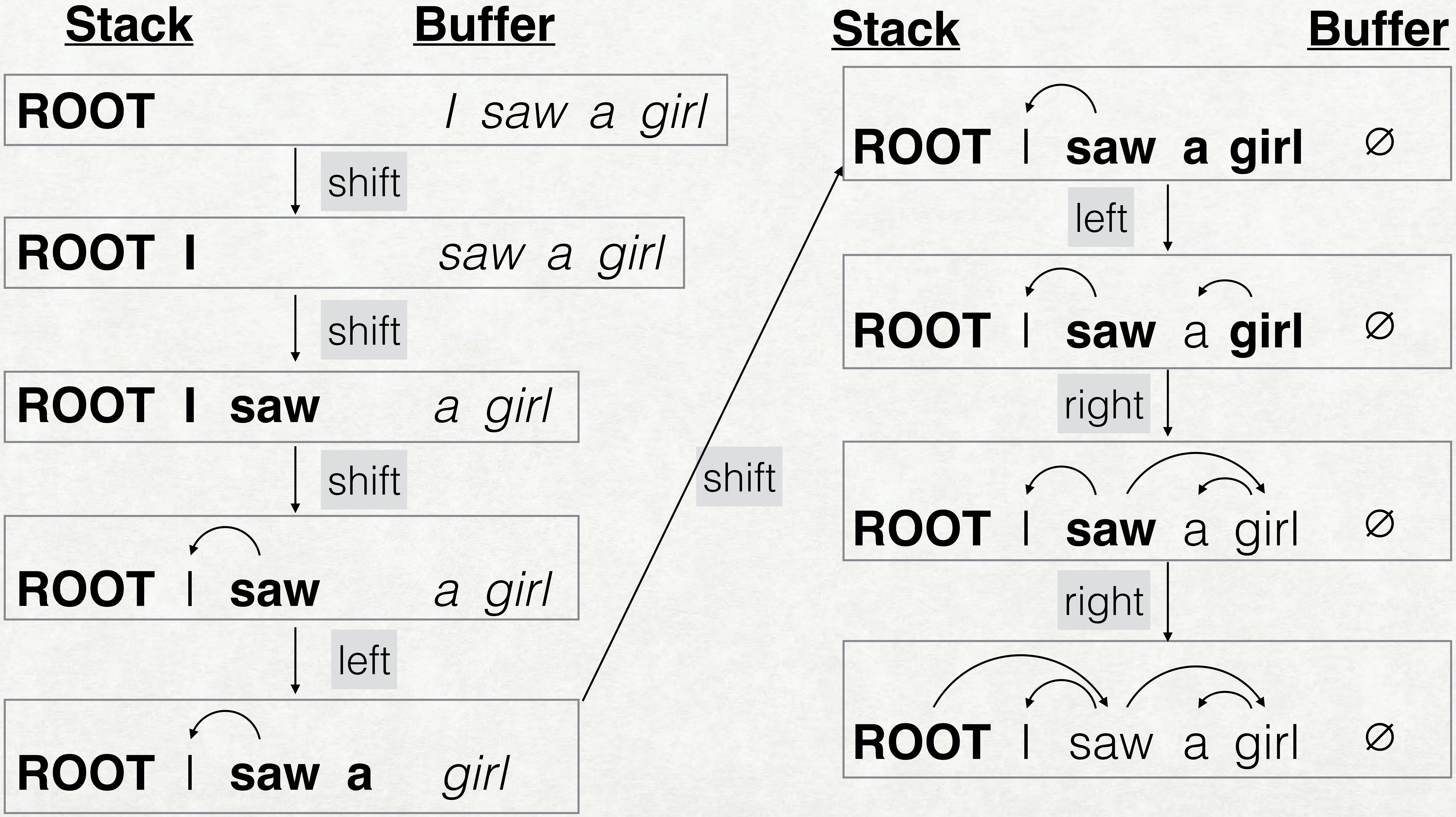
- **Queue**: of unprocessed words
- **Stack**: of partially processed words

At each point choose

- **shift**: move one word from queue to stack
- **reduce left**: top word on stack is head of second word
- **reduce right**: second word on stack is head of top word

Learn how to choose each action with a classifier

SHIFT REDUCE EXAMPLE



CLASSIFICATION FOR SHIFT-REDUCE

Given a configuration

Stack	**Buffer**
ROOT I saw a	*girl*

Which **action** do we choose?

shift

ROOT I saw a girl ∅

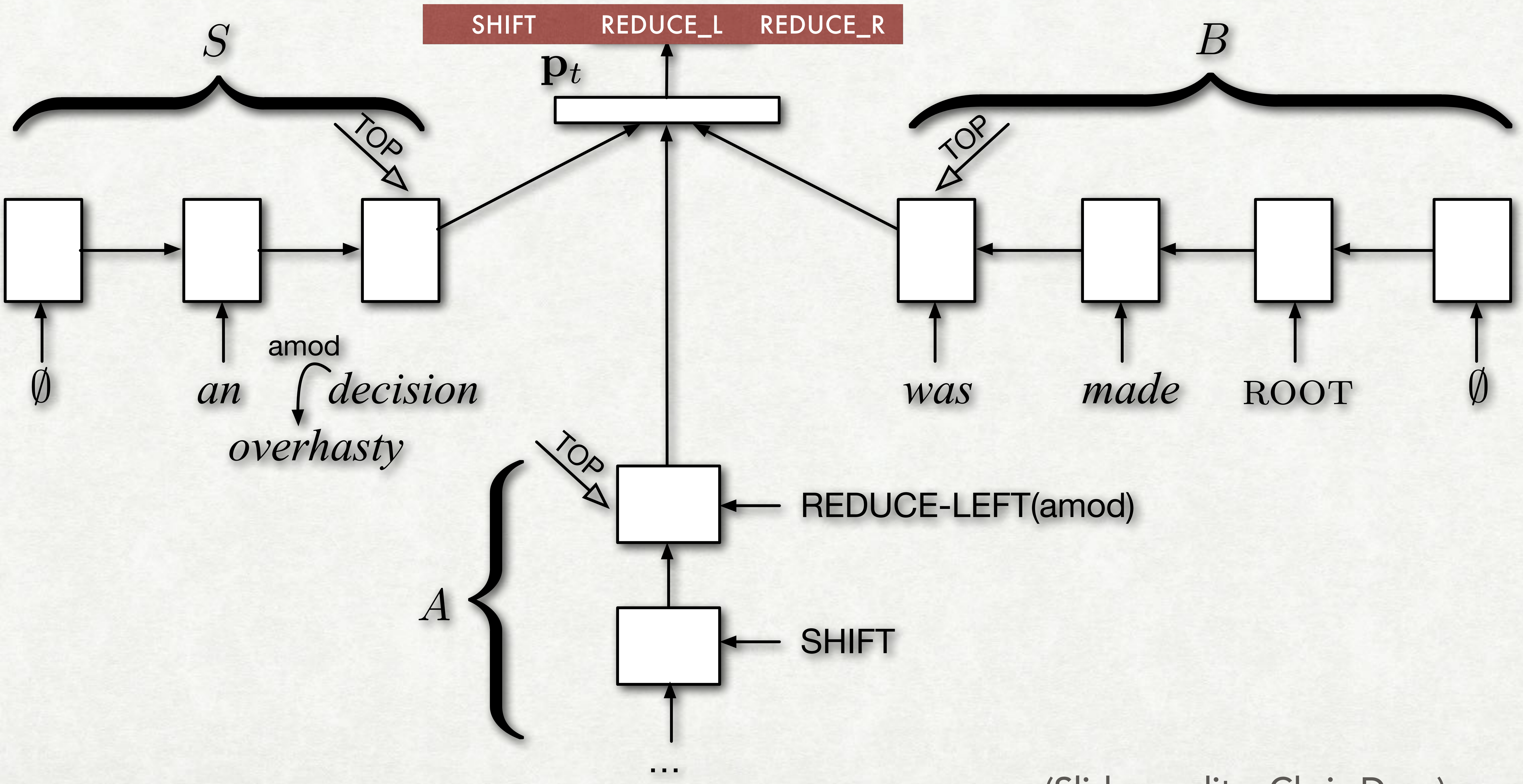
right

ROOT I saw a *girl*

left

ROOT I saw a *girl*

ENCODING STACK CONFIGURATIONS WITH RNNS



(Slide credits: Chris Dyer)

TRANSITION-BASED PARSING

State embeddings

We can embed words, and can embed tree fragments using syntactic compositon

The contents of the buffer are just a sequence of embedded words

which we periodically “shift” from

The contents of the stack is just a sequence of embedded trees

which we periodically pop from and push to

Sequences -> use RNNs to get an encoding!

(Slide credits: Chris Dyer)

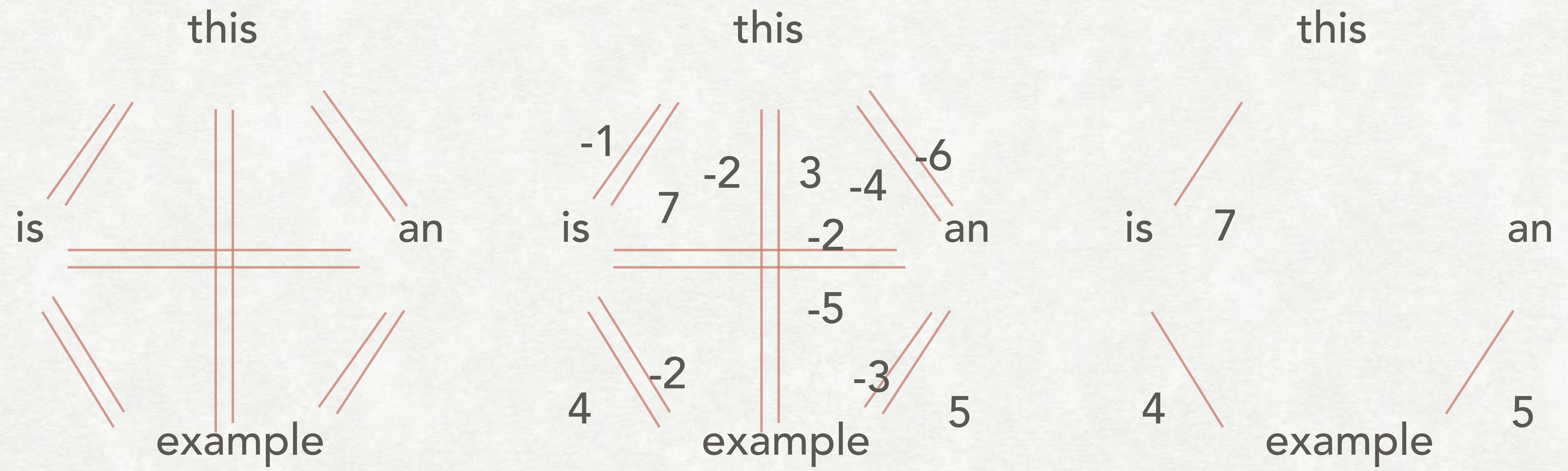
GRAPH-BASED PARSING

(FIRST ORDER) GRAPH-BASED DEPENDENCY PARSING

Express sentence as fully connected directed graph

Score each edge independently

Find maximal spanning tree



GRAPH-BASED VS. TRANSITION BASED

Transition-based

+ Easily condition on infinite tree context (structured prediction)

- Greedy search algorithm causes short-term mistakes

Graph-based

+ Can find exact best global solution via DP algorithm

- Have to make local independence assumptions

CHU-LIU-EDMONDS (CHU AND LIU 1965, EDMONDS 1967)

We have a graph and want to find its spanning tree

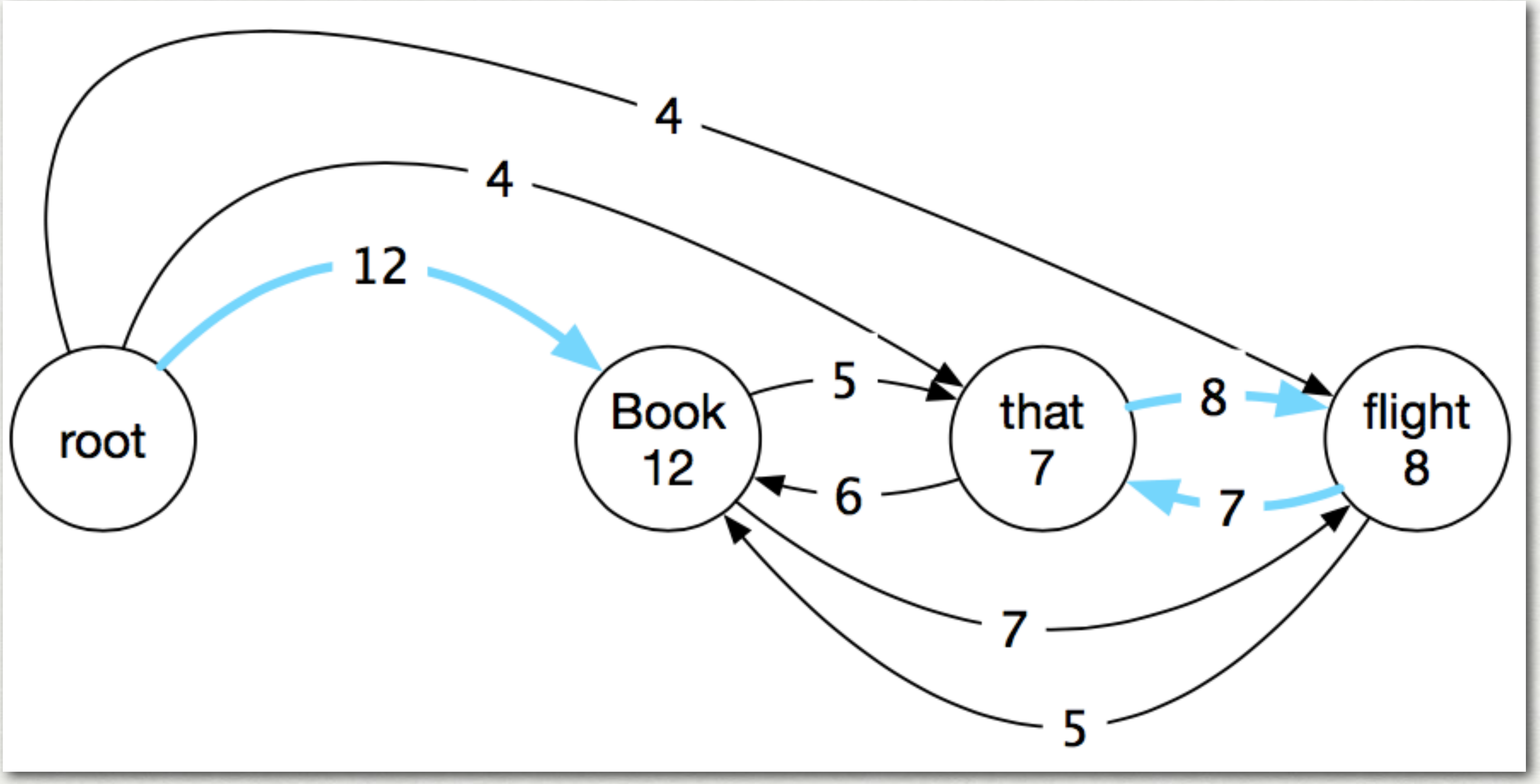
Greedily select the best incoming edge to each node (and subtract its score from all incoming edges)

If there are cycles, select a cycle and **contract** it into a single node

Recursively call the algorithm on the graph with the contracted node

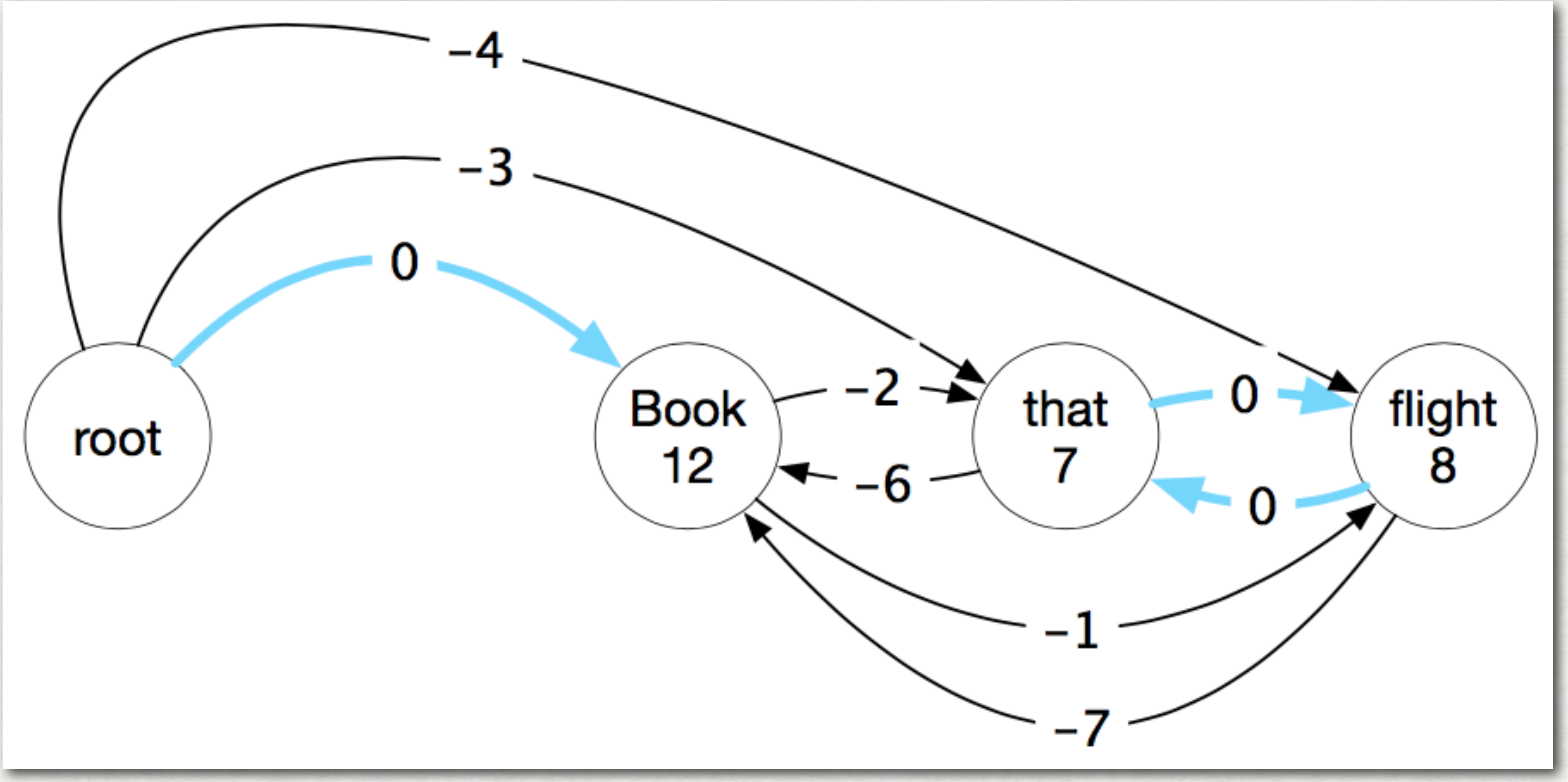
Expand the contracted node, deleting an edge appropriately

CHU-LIU-EDMONDS (1): FIND THE BEST INCOMING



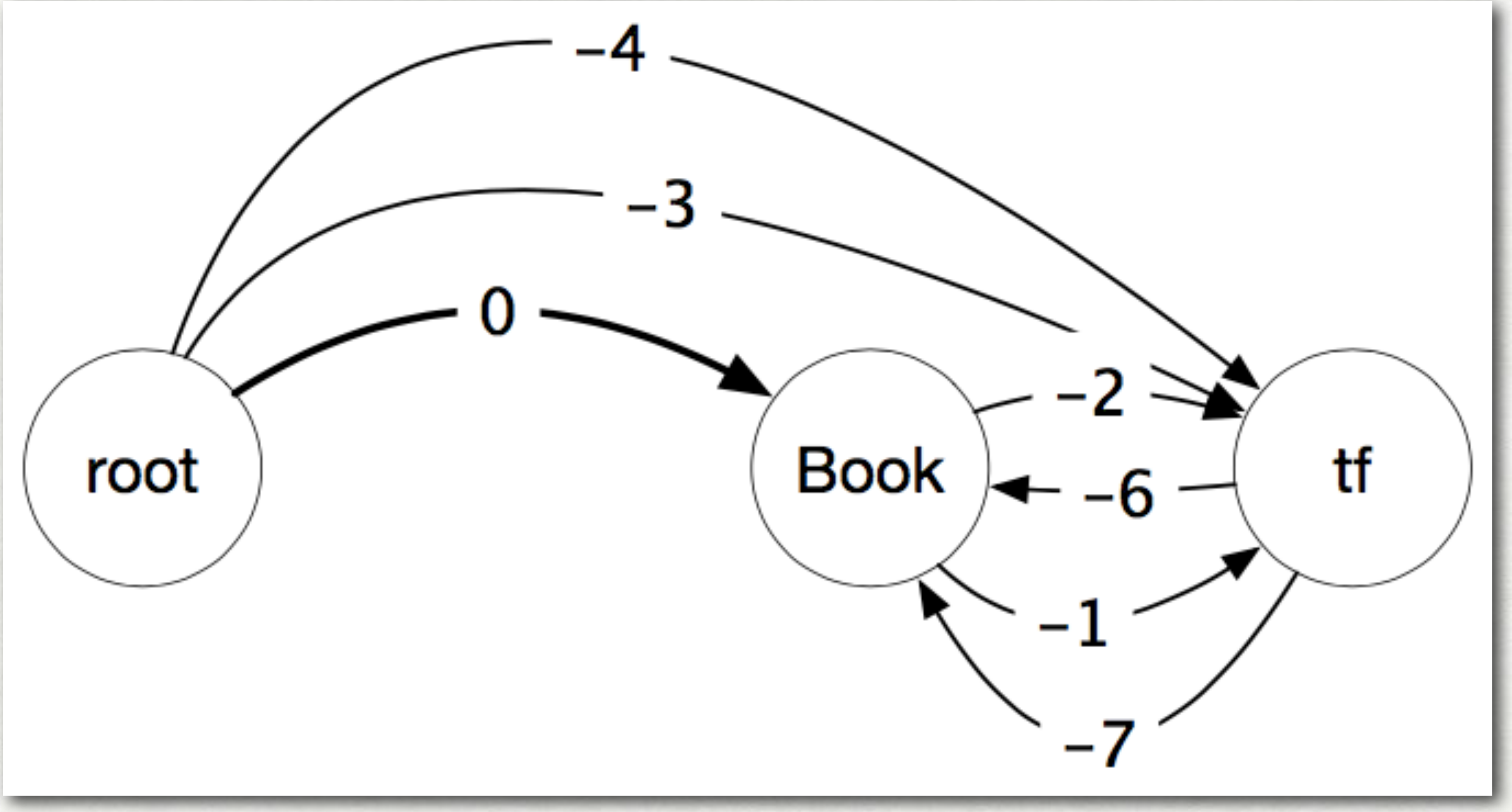
(Figure Credit: Jurafsky and Martin)

CHU-LIU-EDMONDS (2): SUBTRACT THE MAX FOR EACH



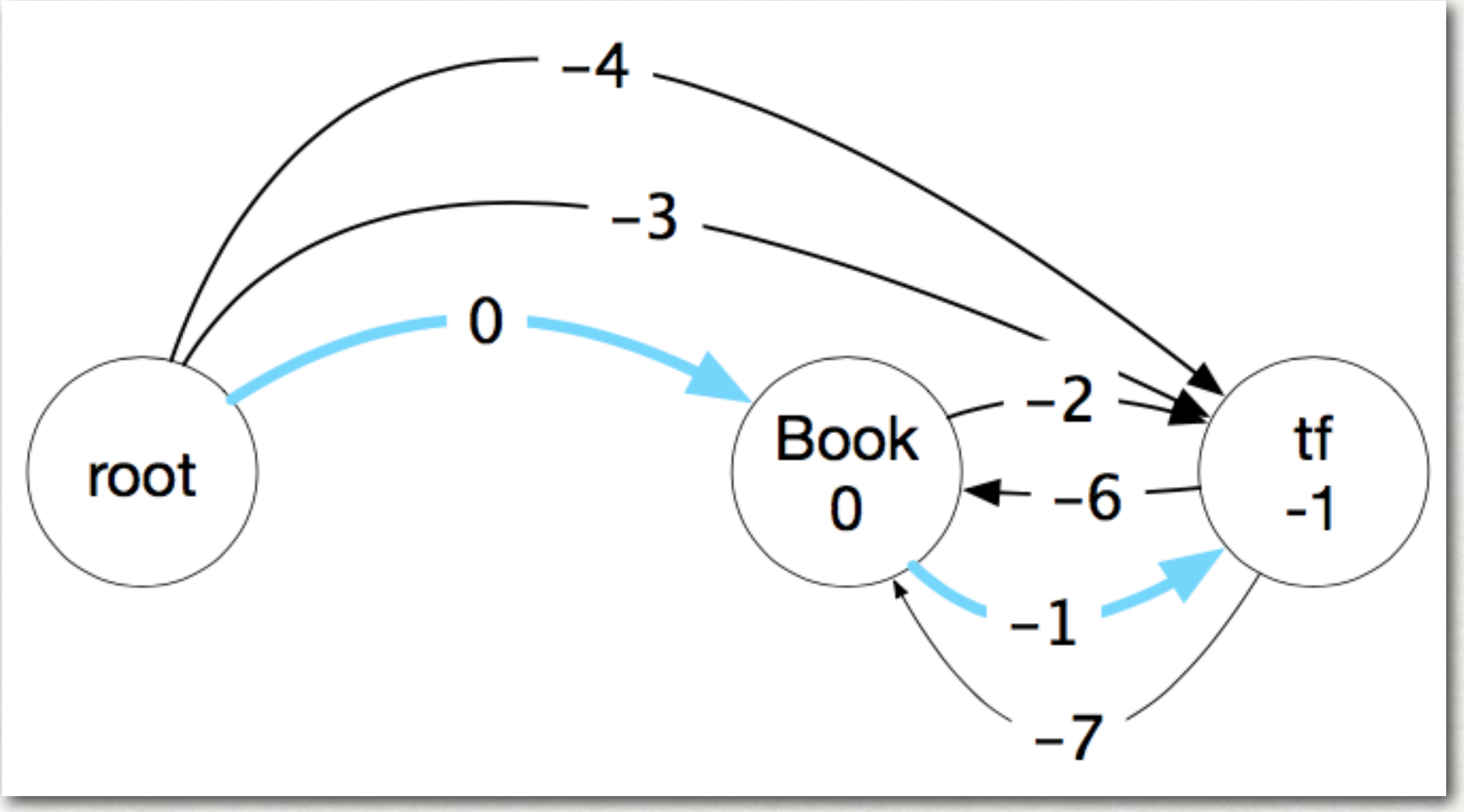
(Figure Credit: Jurafsky and Martin)

CHU-LIU-EDMONDS (3): CONTRACT A NODE



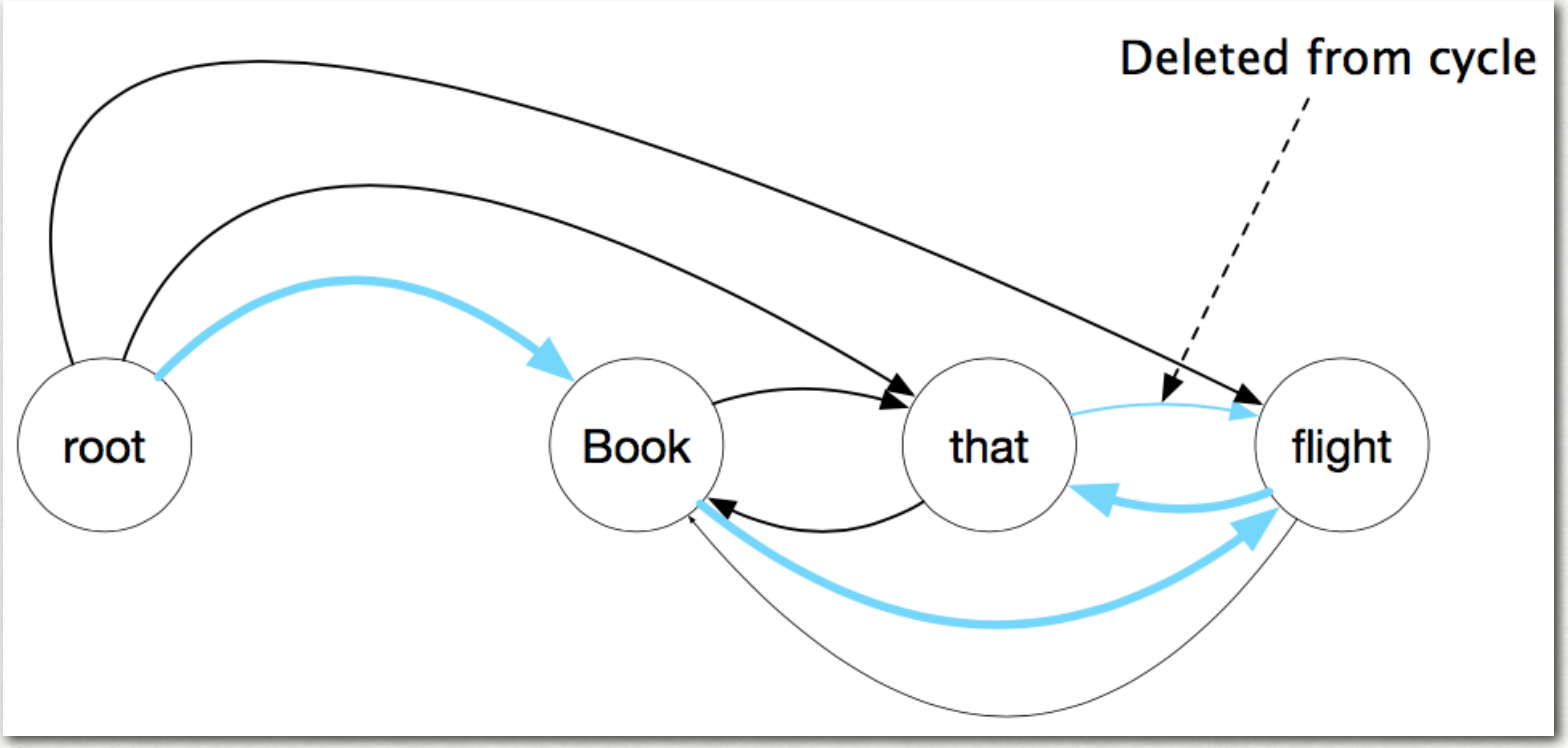
(Figure Credit: Jurafsky and Martin)

CHU-LIU-EDMONDS (4): RECURSIVELY CALL ALGORITHM



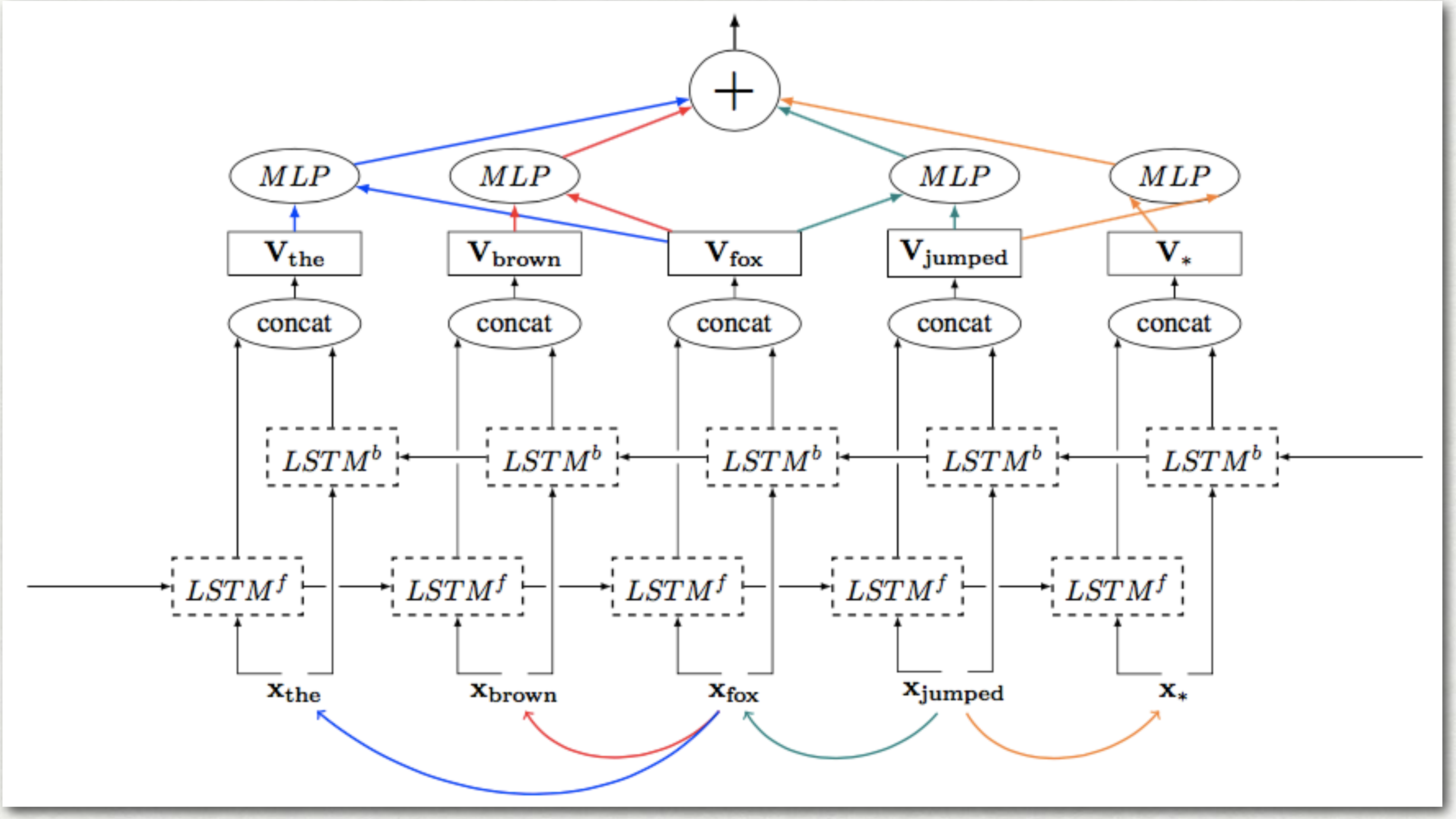
(Figure Credit: Jurafsky and Martin)

CHU-LIU-EDMONDS (5): EXPAND NODES AND DELETE EDGE



(Figure Credit: Jurafsky and Martin)

SEQUENCE MODEL FEATURE EXTRACTORS (KIPPERWASSER AND GOLDBERG 2016)



BIAFFINE CLASSIFIER (DOZAT AND MANNING 2017)

$$\mathbf{h}_i^{(arc\text{-}dep)} = \mathrm{MLP}^{(arc\text{-}dep)}(\mathbf{r}_i)$$
$$\mathbf{h}_j^{(arc\text{-}head)} = \mathrm{MLP}^{(arc\text{-}head)}(\mathbf{r}_j)$$
$$\mathbf{s}_i^{(arc)} = H^{(arc\text{-}head)} U^{(1)} \mathbf{h}_i^{(arc\text{-}dep)} + H^{(arc\text{-}head)} \mathbf{u}^{(2)}$$

Learn specific representations for head/dependent for each word

Calculate score of each arc

Just optimize the likelihood of the parent, no structured training

This is a local model, with global decoding using MST at the end

Best results (with careful parameter tuning) on universal dependencies parsing task

MULTILINGUAL DEPENDENCY PARSING

DIFFICULTY IN MULTILINGUAL DEPENDENCY PARSING

Syntactic analysis is a particularly hard multilingual task

It is on the global level, not just word-by-word level

Syntax varies widely across different languages

EXAMPLE IMPROVEMENT 1: ORDER-INSENSITIVE ENCODERS

Standard cross-lingual transfer can fail with large word order differences between source and target

Change model structure to be order-insensitive to avoid over-fitting to source

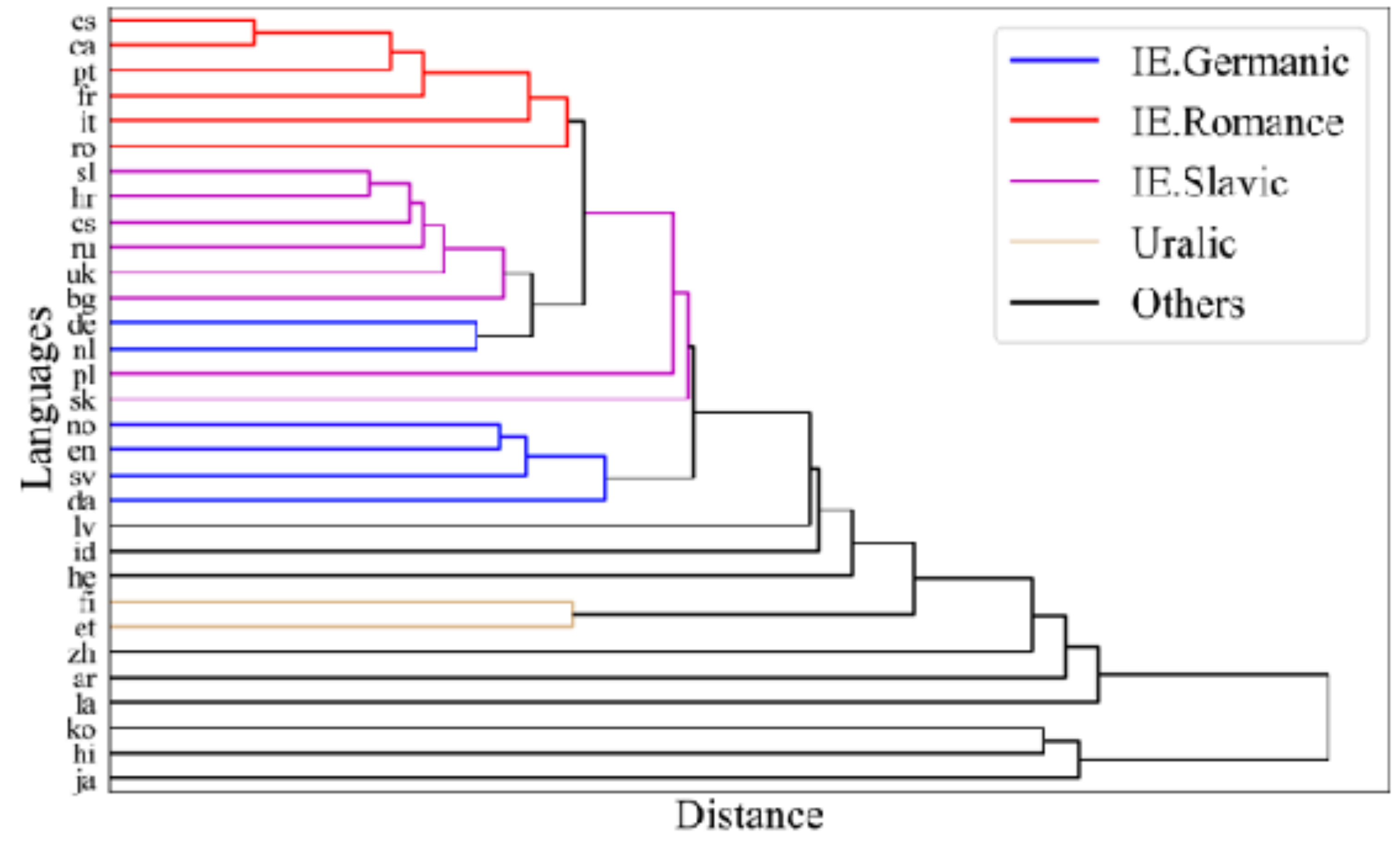


Figure 1: Hierarchical clustering (with the Nearest Point Algorithm) dendrogram of the languages by their word-ordering vectors.

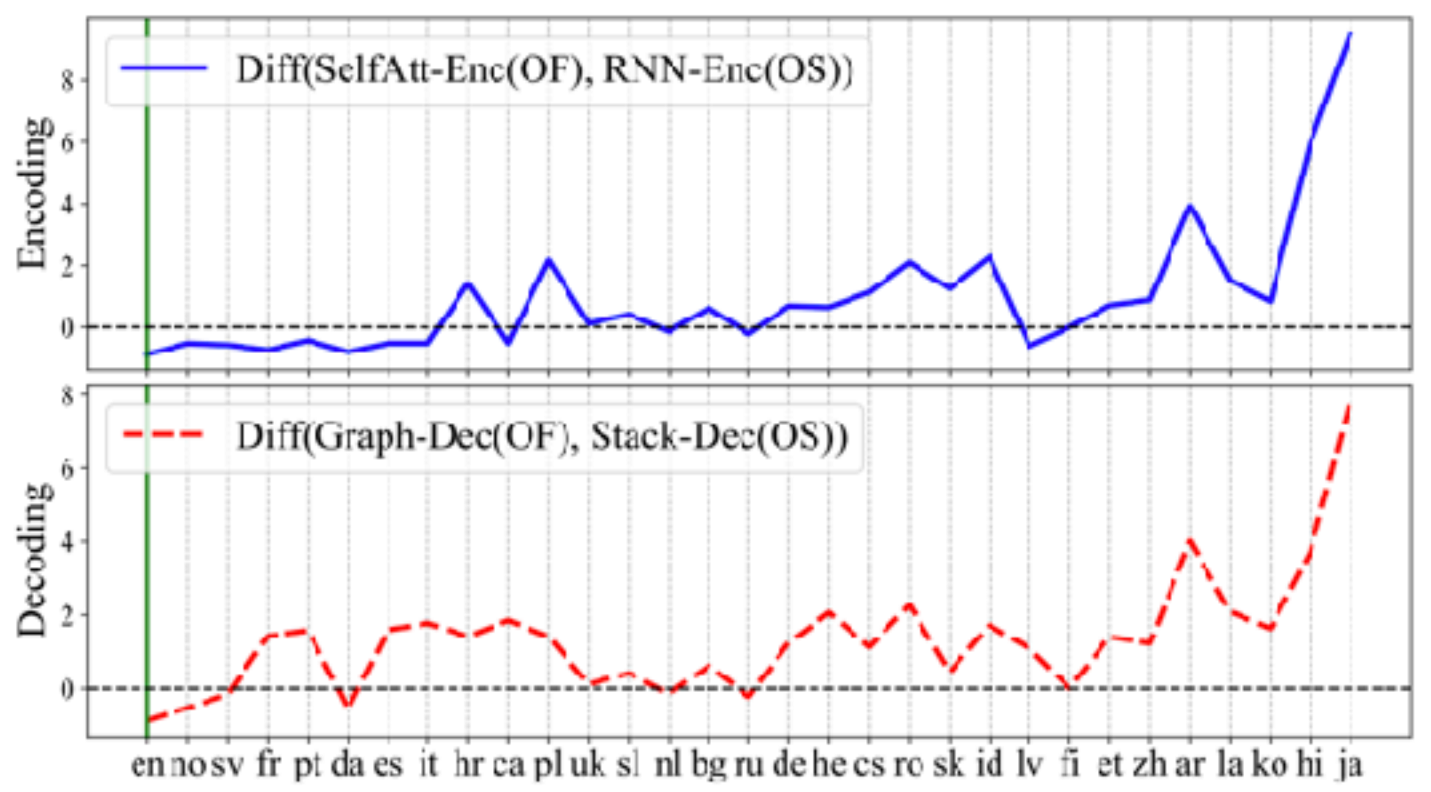
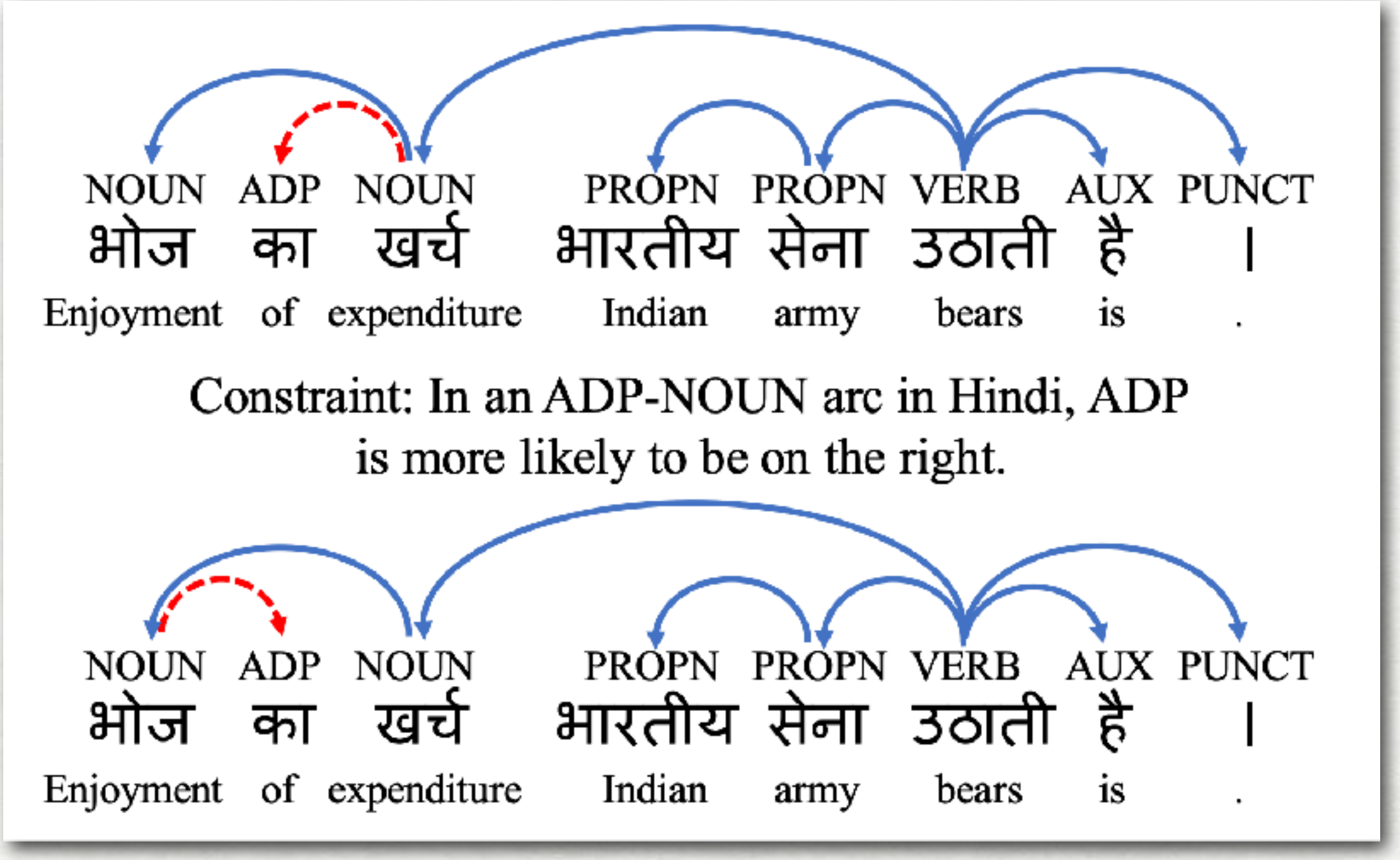


Figure 2: Evaluation score differences between Order-Free (OF) and Order Sensitive (OS) modules. We show results of both encoder (blue solid curve) and decoder (dashed red curve). Languages are sorted by their word-ordering distances to English from left to right. The position of English is marked with a green bar.

Ahmad, Wasi Uddin, et al. "On difficulties of cross-lingual transfer with order differences: A case study on dependency parsing." NAACL 2019.

EXAMPLE IMPROVEMENT 2: LINGUISTICALLY INFORMED CONSTRAINTS

Add constraints based on a-priori knowledge of the language structure



Meng, Tao, Nanyun Peng, and Kai-Wei Chang. "Target language-aware constrained inference for cross-lingual dependency parsing." EMNLP 2019.

NEXT CLASS

Also, check this cool pytorch library: PyTorch-Struct
http://nlp.seas.harvard.edu/pytorch-struct/index.html

Next class: Lexical Semantics