# Synthetic Data Made to Order: The Case of Parsing

Dingquan Wang and Jason Eisner
Department of Computer Science
Johns Hopkins University

# Acknowledgement

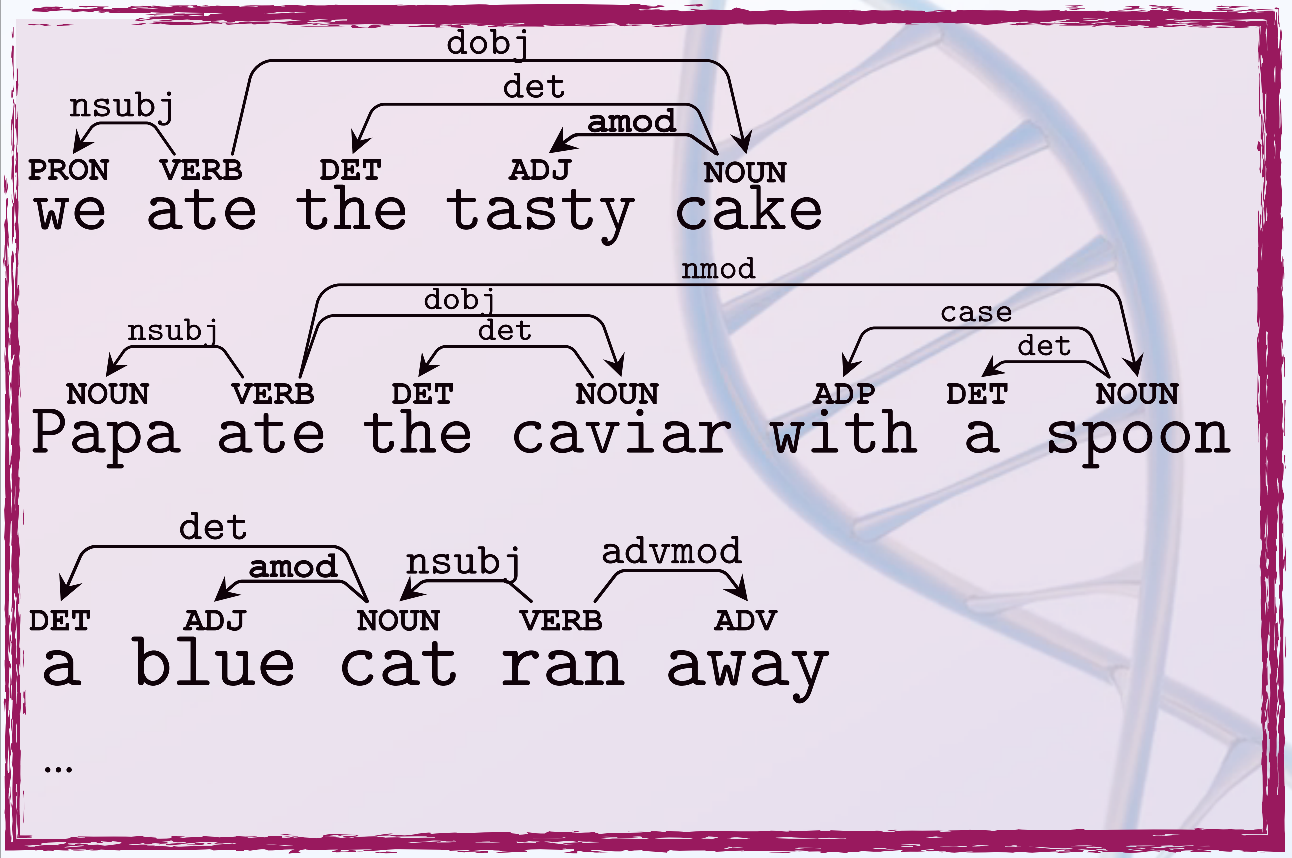
- Universal Dependencies
  - Now has **122** treebanks, **71** languages
  - Empower the supervised methods to process these languages
  - Help analyze novel languages

**Transfer Parsing**

JOHNS HOPKINS
UNIVERSITY

# Transfer Parsing


## English Treebank




## French Corpus

**Ma mère s'appelle Emilie Summer**

**Lundi, je retourne à l’école**

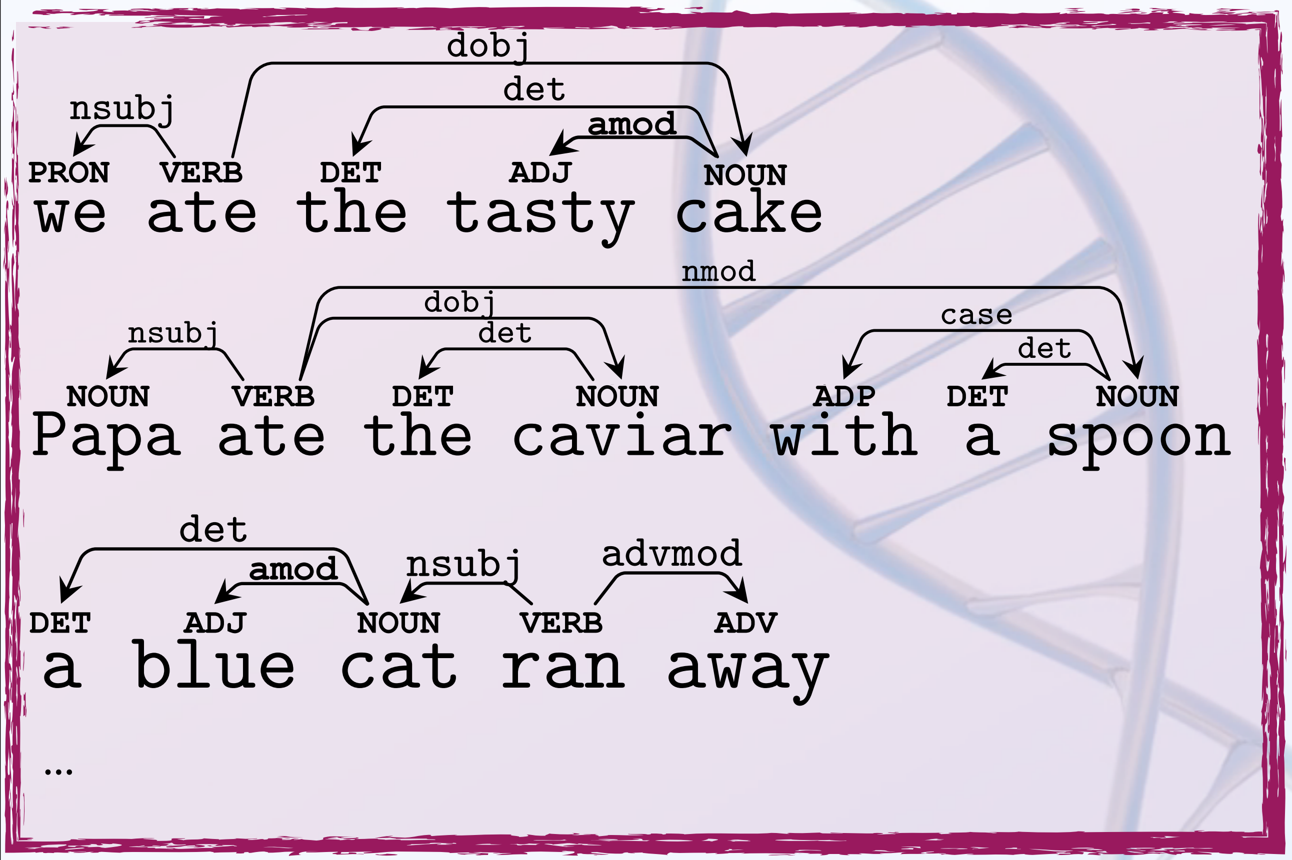
**C’est ma meilleure amie**

**J’aime beaucoup l’école**

…

# Transfer Parsing


## English Treebank




## French Corpus

Ma mère s'appelle Emilie Summer

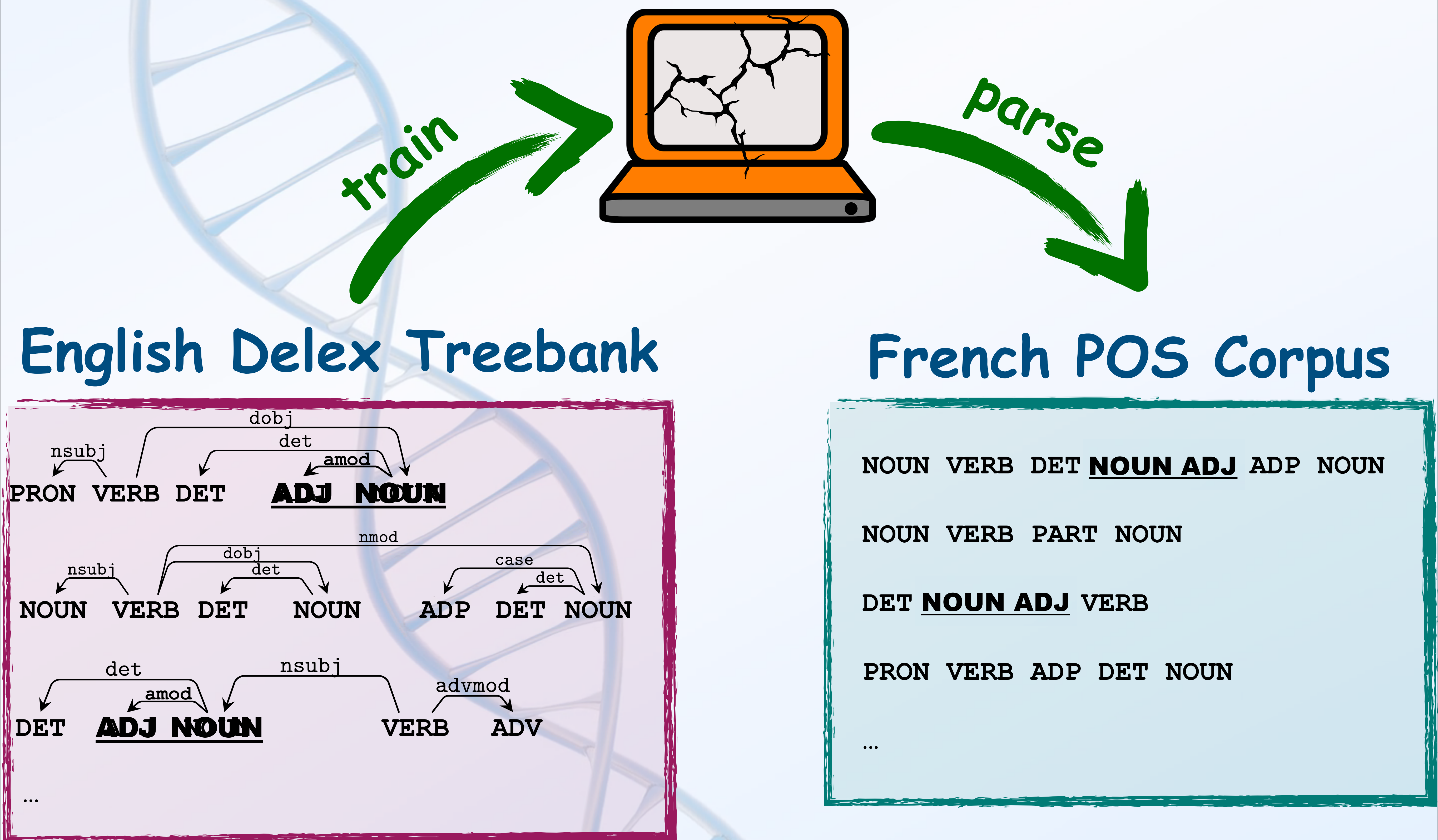
Lundi, je retourne à l’école

C’est ma meilleure amie

J’aime beaucoup l’école

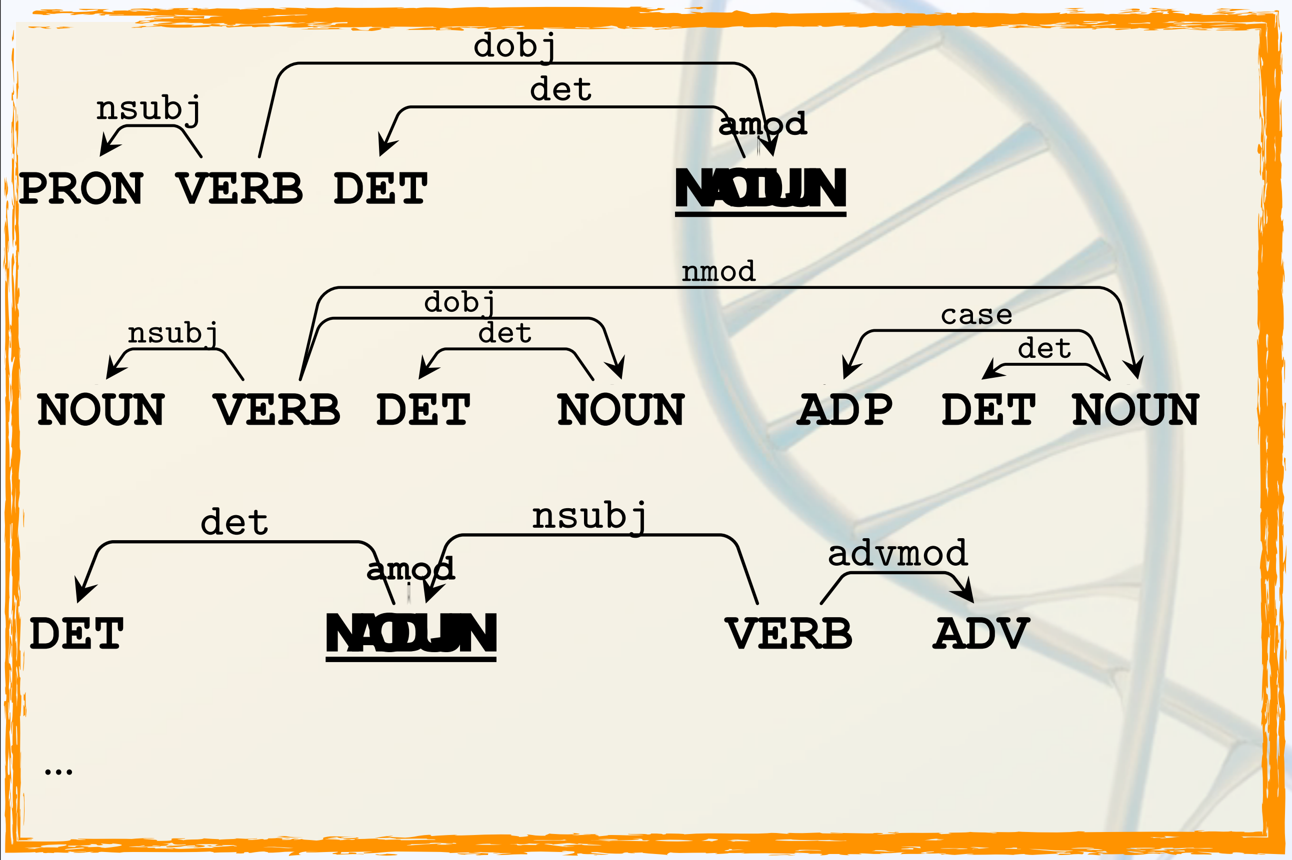
…

# Delexicalized Transfer Parsing



# Delexicalized Transfer Parsing


## English Delex Treebank




## French POS Corpus

NOUN VERB DET **NOUN ADJ** ADP NOUN

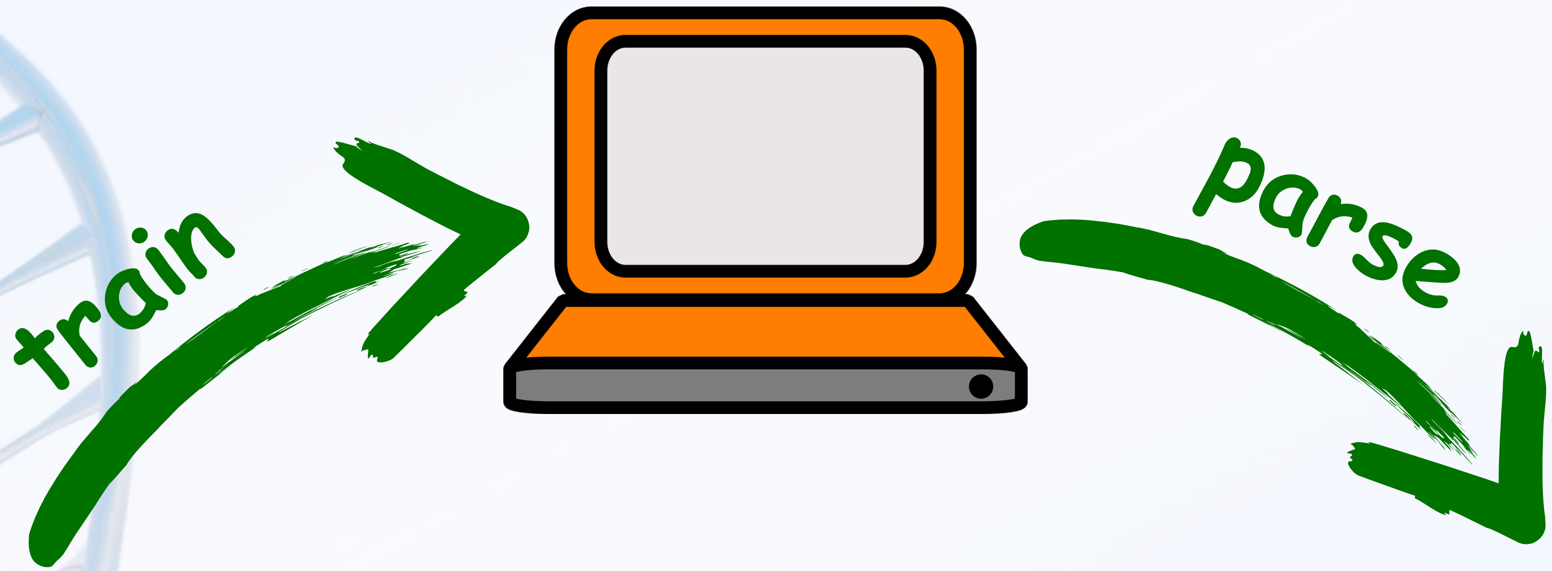
NOUN VERB PART NOUN

DET **NOUN ADJ** VERB

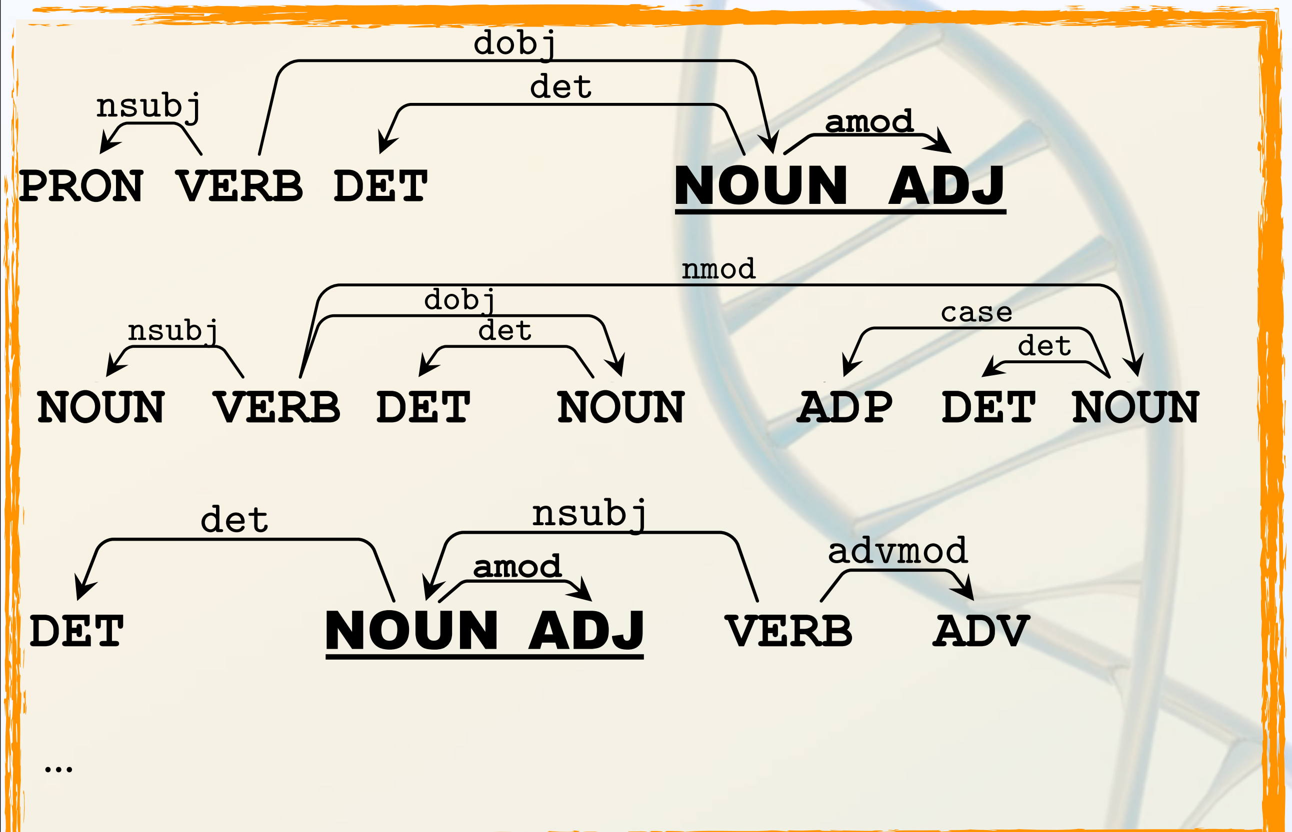
PRON VERB ADP DET NOUN

…

# Delexicalized Transfer Parsing




## English' Delex Treebank




## French POS Corpus

NOUN VERB DET **NOUN ADJ** ADP NOUN

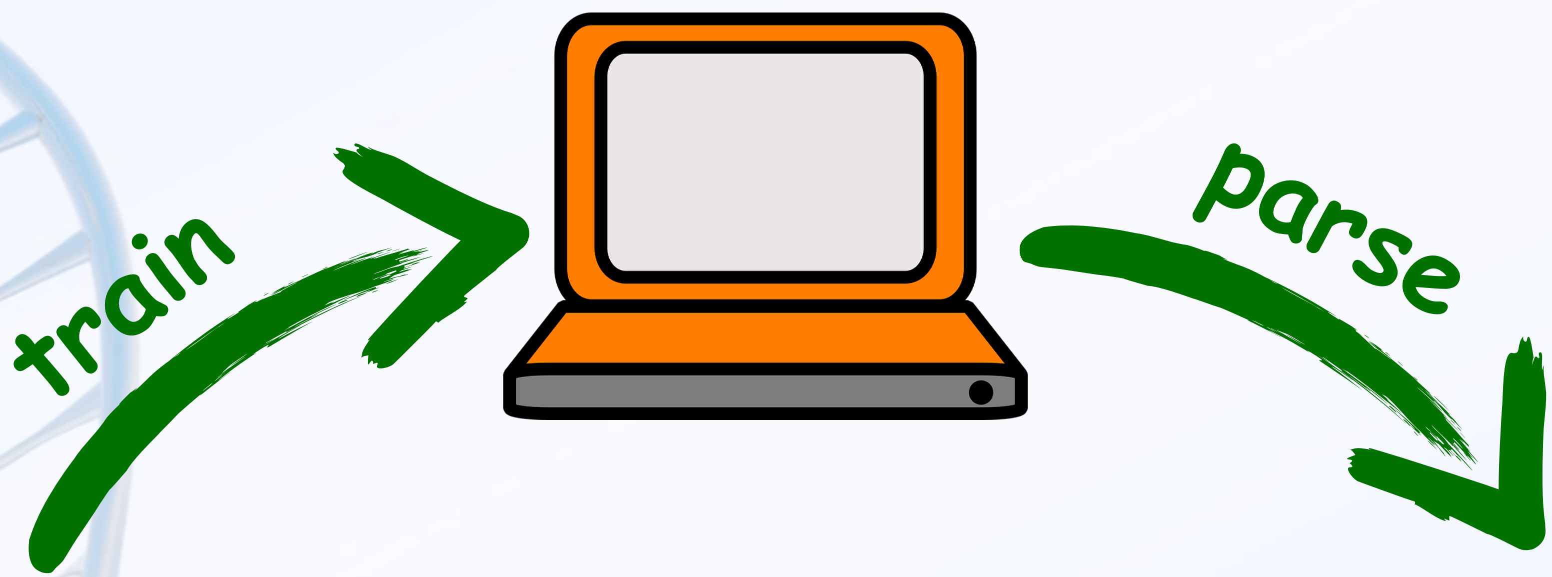
NOUN VERB PART NOUN

DET **NOUN ADJ** VERB

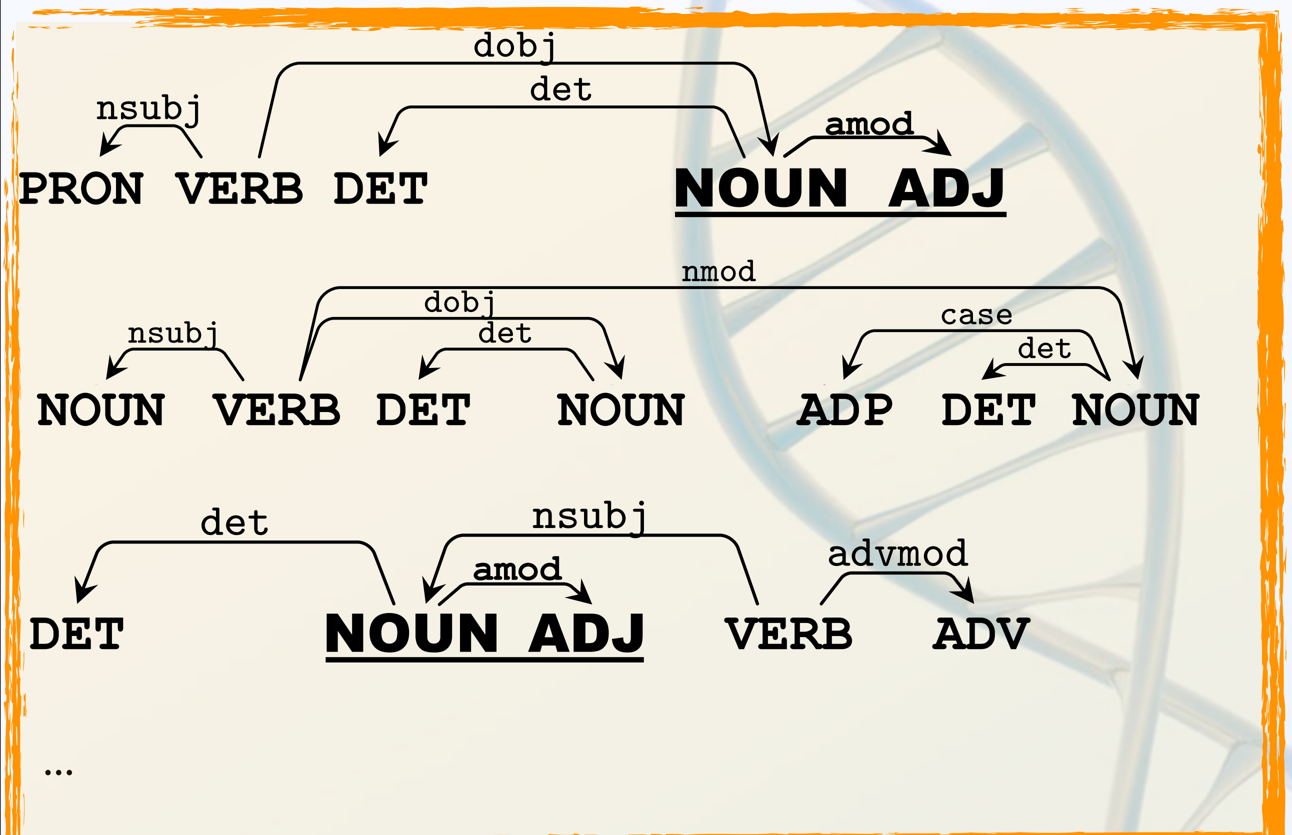
PRON VERB ADP DET NOUN

…

# Delexicalized Transfer Parsing




## English' Delex Treebank




## French POS Corpus

NOUN VERB DET **NOUN ADJ** ADP NOUN

NOUN VERB PART NOUN

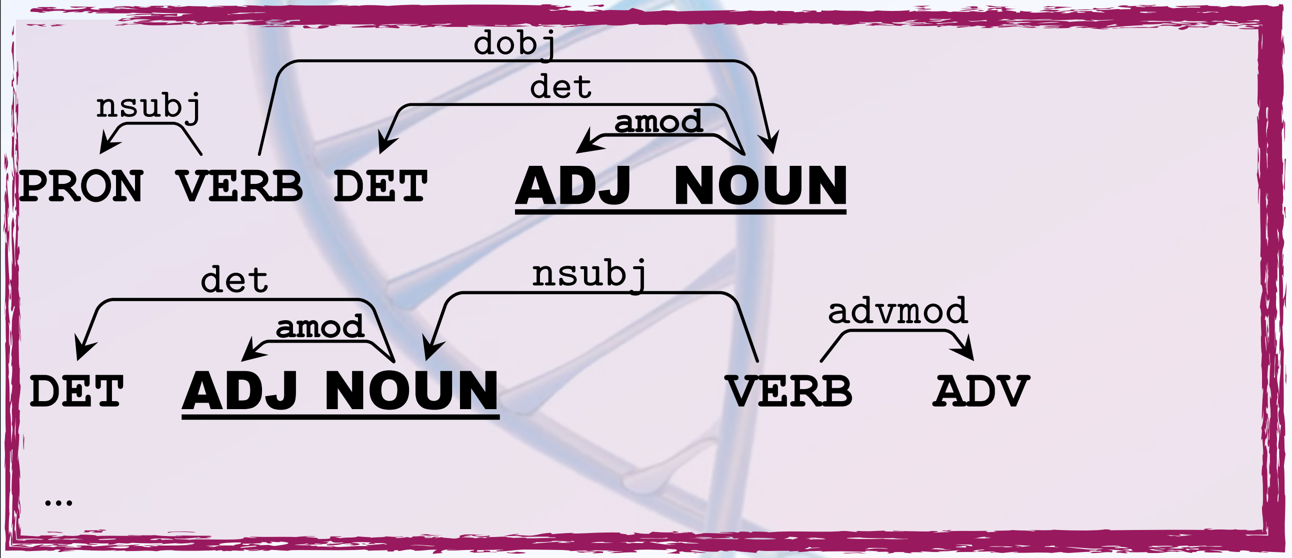
DET **NOUN ADJ** VERB

PRON VERB ADP DET NOUN

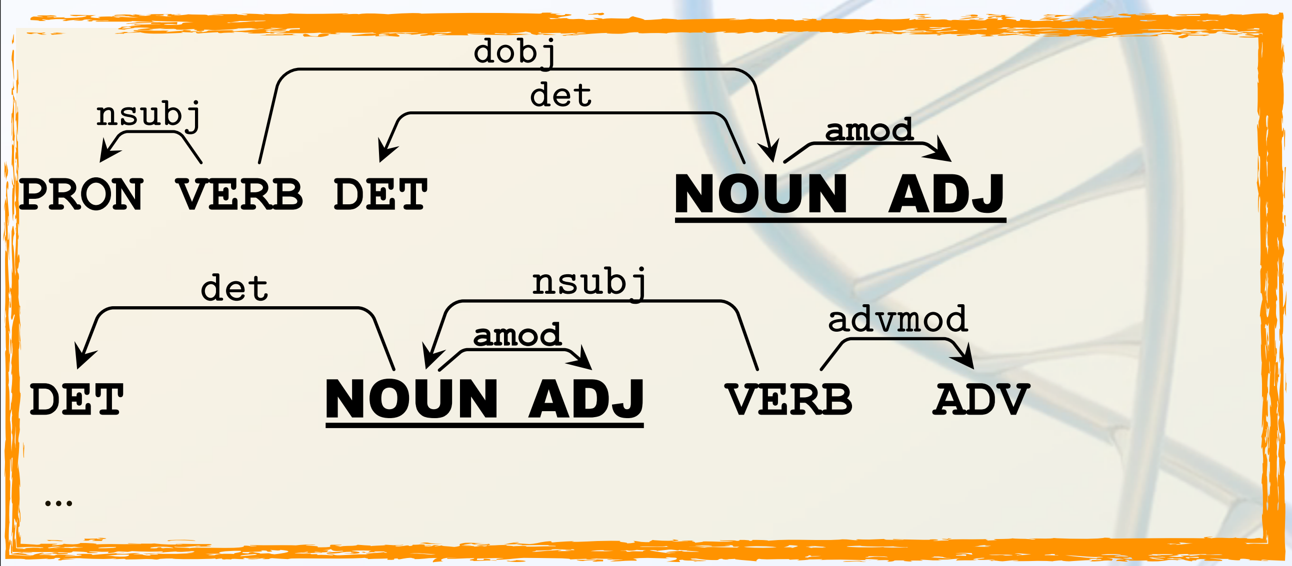
…

# Improve the surface similarity

## English




## English'




## French POS Corpus

NOUN VERB DET **NOUN ADJ** ADP NOUN

NOUN VERB PART NOUN

DET **NOUN ADJ** VERB

PRON VERB ADP DET NOUN

…

# Improve the surface similarity

## English

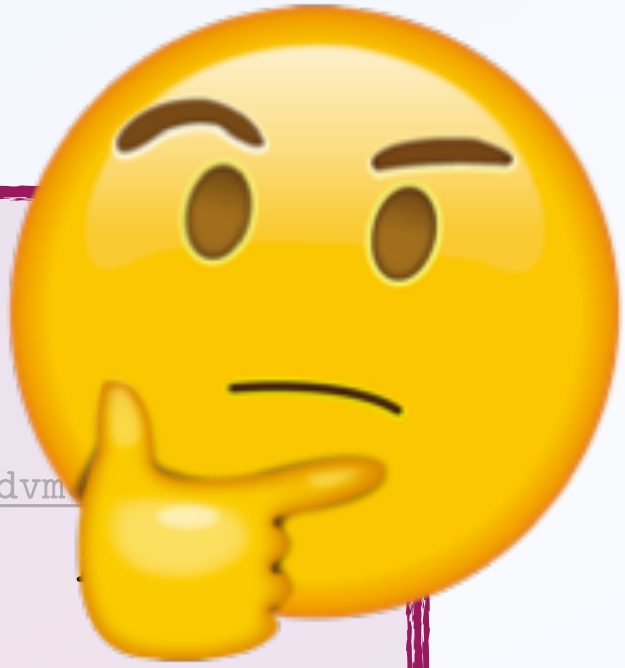
dobj det nsubj amod

PRON VERB DET **ADJ NOUN**

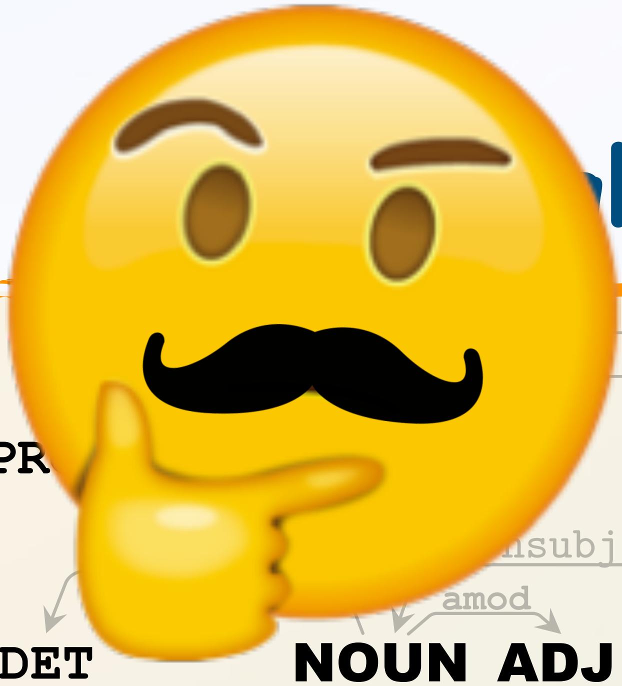
det nsubj amod advm

DET **ADJ NOUN** VERB

…



## lish'



amod

PR **NOUN ADJ**

nsubj amod advmod

DET **NOUN ADJ** VERB ADV

…

## French POS Corpus

NOUN VERB DET **NOUN ADJ** ADP NOUN

NOUN VERB PART NO

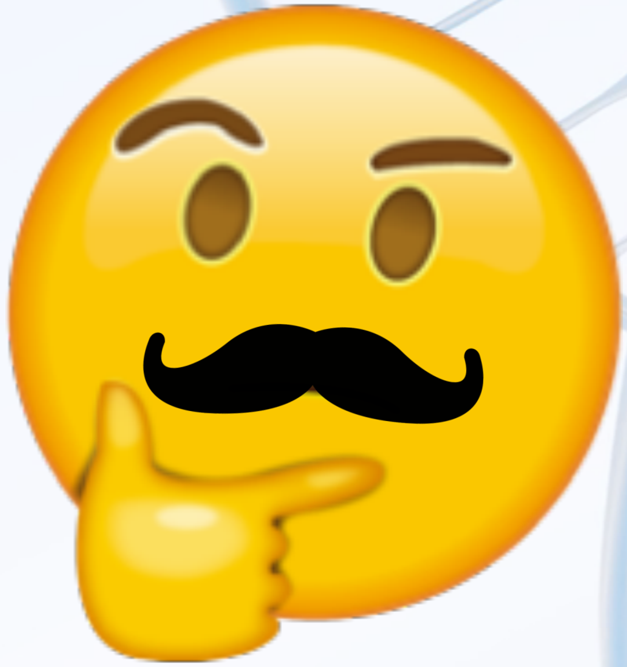
DET **NOUN ADJ** V

PRON VERB ADP I

…

# Improve the surface similarity

Source'



Target POS corpus

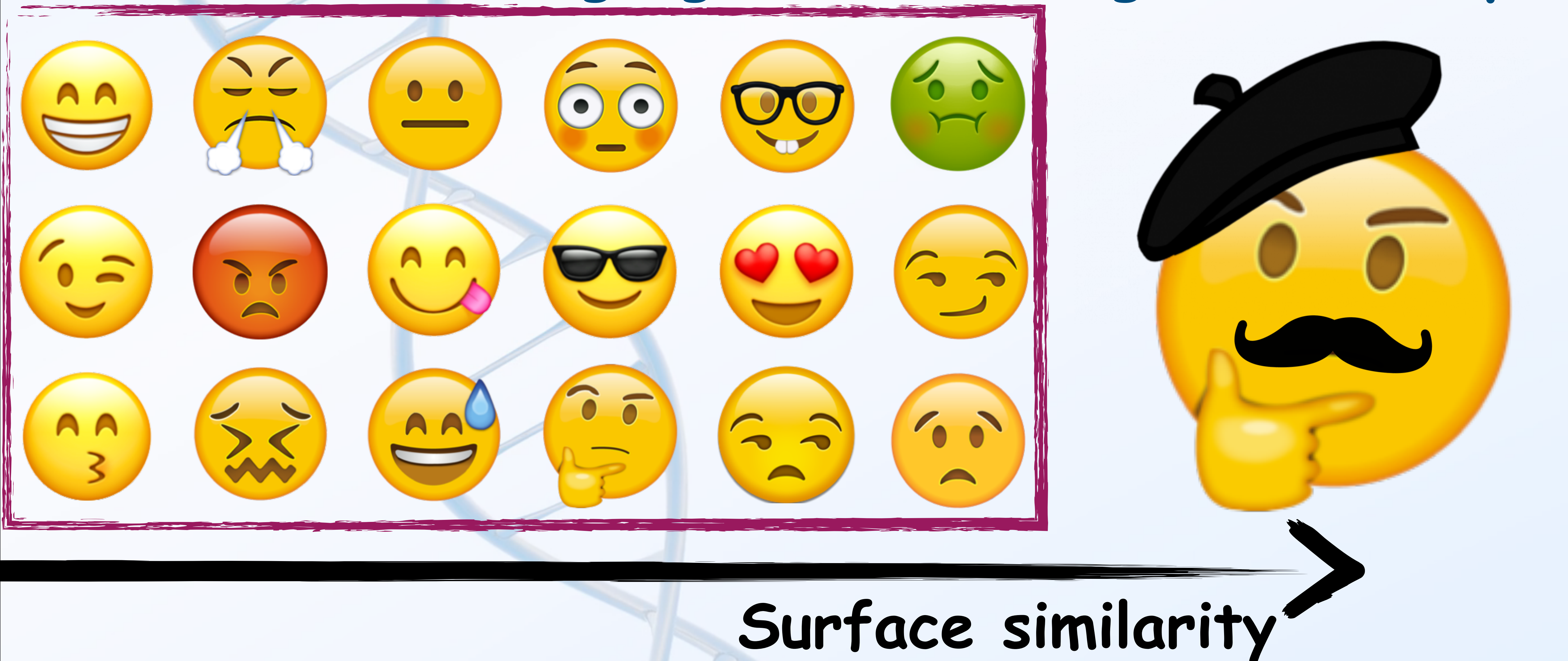
Surface similarity

Transfer Parsing accuracy?

JOHNS HOPKINS UNIVERSITY

# Single-Source Selection

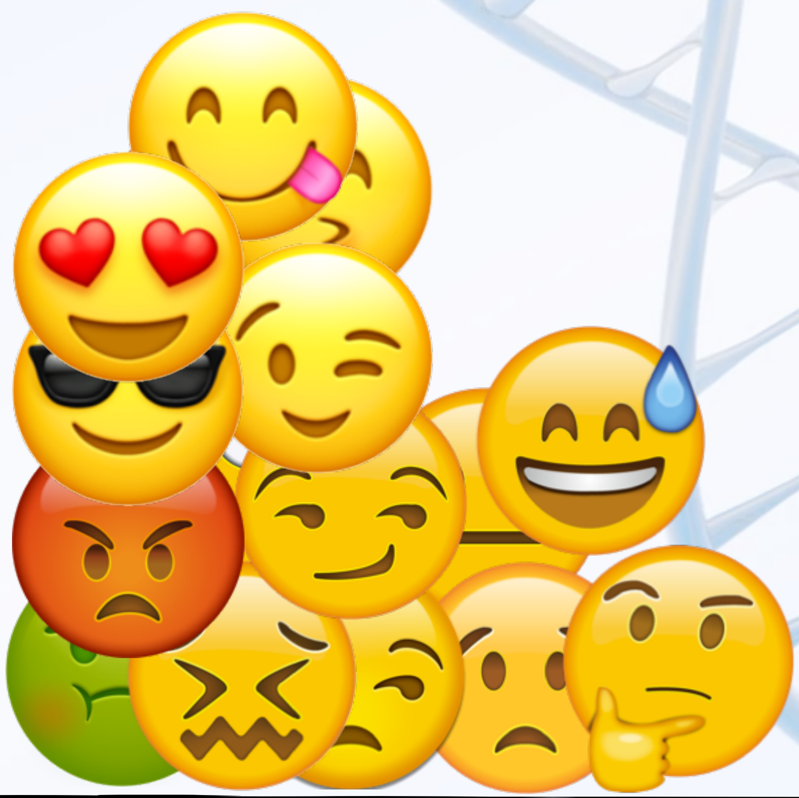
Source languages    Target POS corpus



# Single-Source Selection

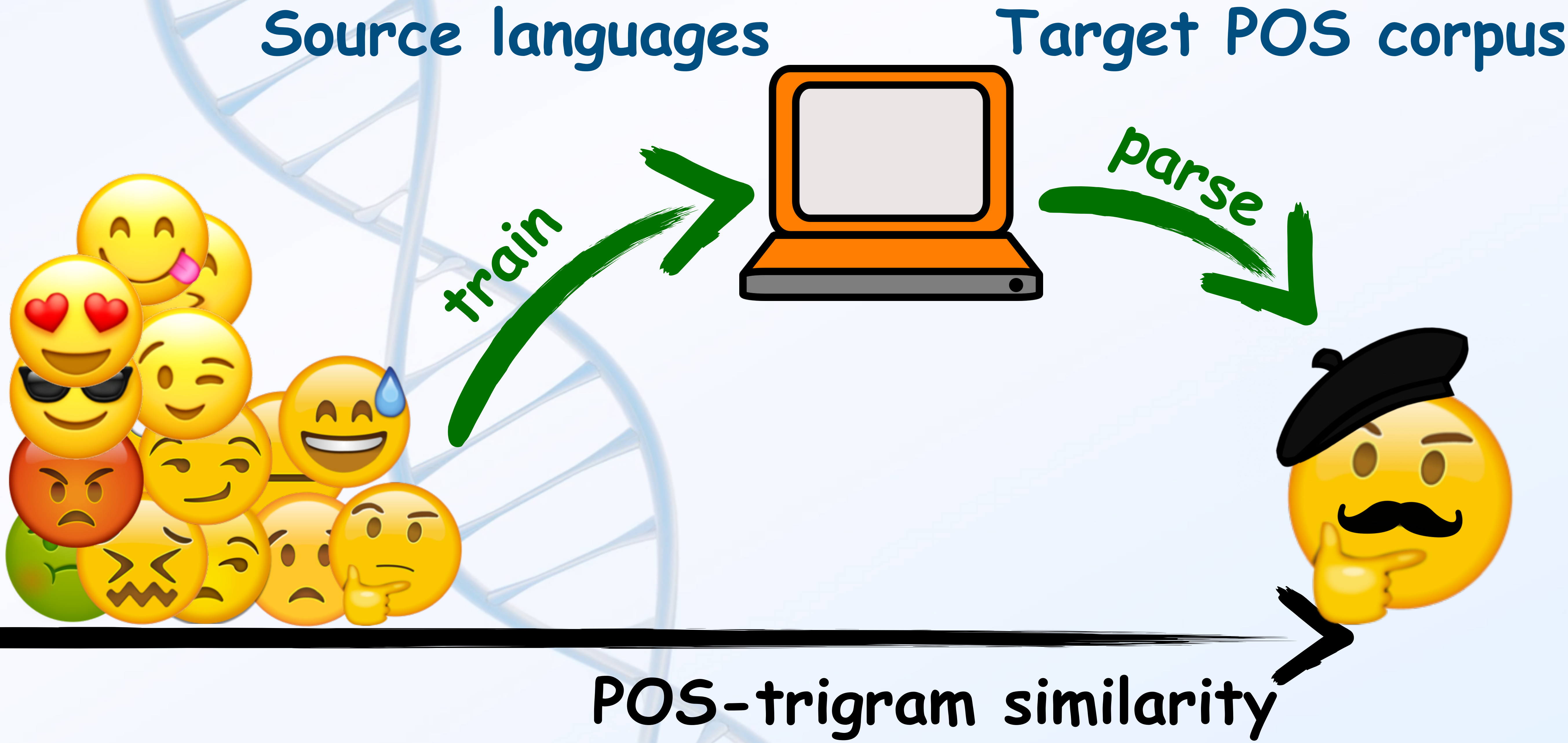
Source languages

Target POS corpus



POS-trigram similarity

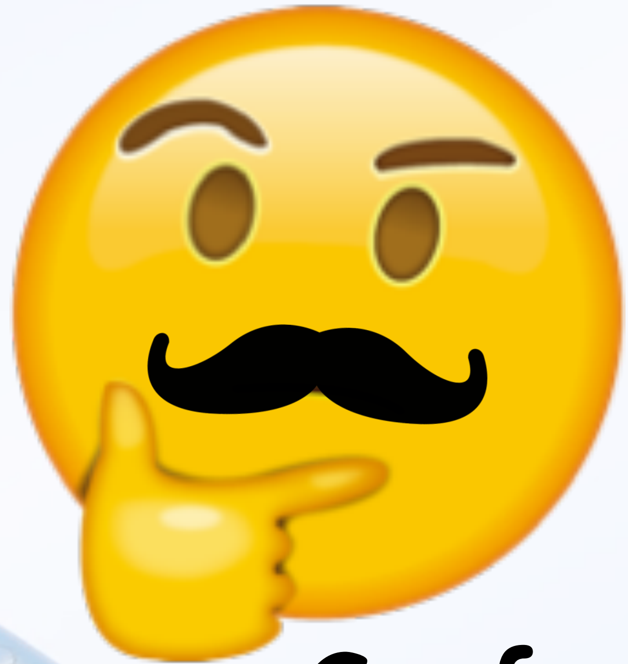
# Single-Source Selection



# Synthetic Data

Source

Source'



Target
POS corpus

Surface similarity

dobj
nsubj det amod
PRON VERB DET **ADJ NOUN**

det amod nsubj advmod
DET **ADJ NOUN** VERB ADV

...

dobj
nsubj det amod
PRON VERB DET **NOUN ADJ**

nsubj
det amod advmod
DET **NOUN ADJ** VERB ADV

...

NOUN VERB DET **NOUN ADJ**

NOUN VERB PART NOUN

DET **NOUN ADJ** VERB

PRON VERB ADP DET NOUN

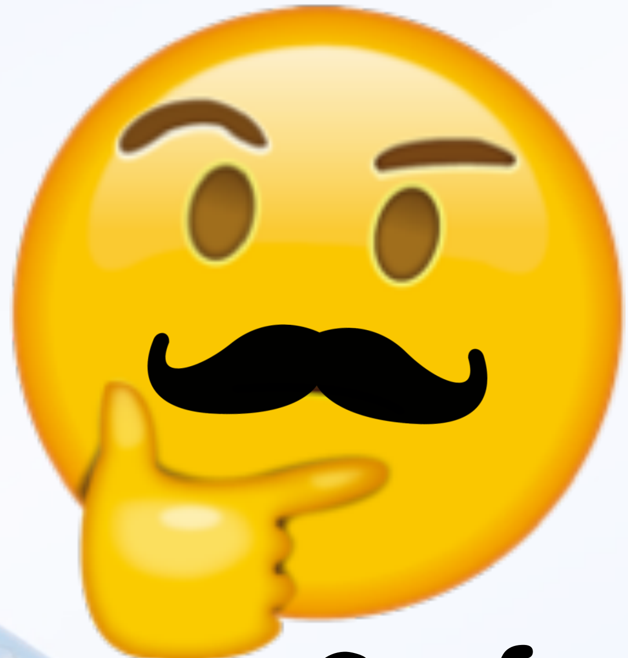
...

JOHNS HOPKINS
UNIVERSITY

# Synthetic Data

Source

Synthetic Source'



Target
POS corpus

Surface similarity

dobj
nsubj det amod
PRON VERB DET **ADJ NOUN**

det amod nsubj advmod
DET **ADJ NOUN** VERB ADV

...

dobj
nsubj det amod
PRON VERB DET **NOUN ADJ**

nsubj
det amod advmod
DET **NOUN ADJ** VERB ADV

...

NOUN VERB DET **NOUN ADJ**

NOUN VERB PART NOUN

DET **NOUN ADJ** VERB

PRON VERB ADP DET NOUN

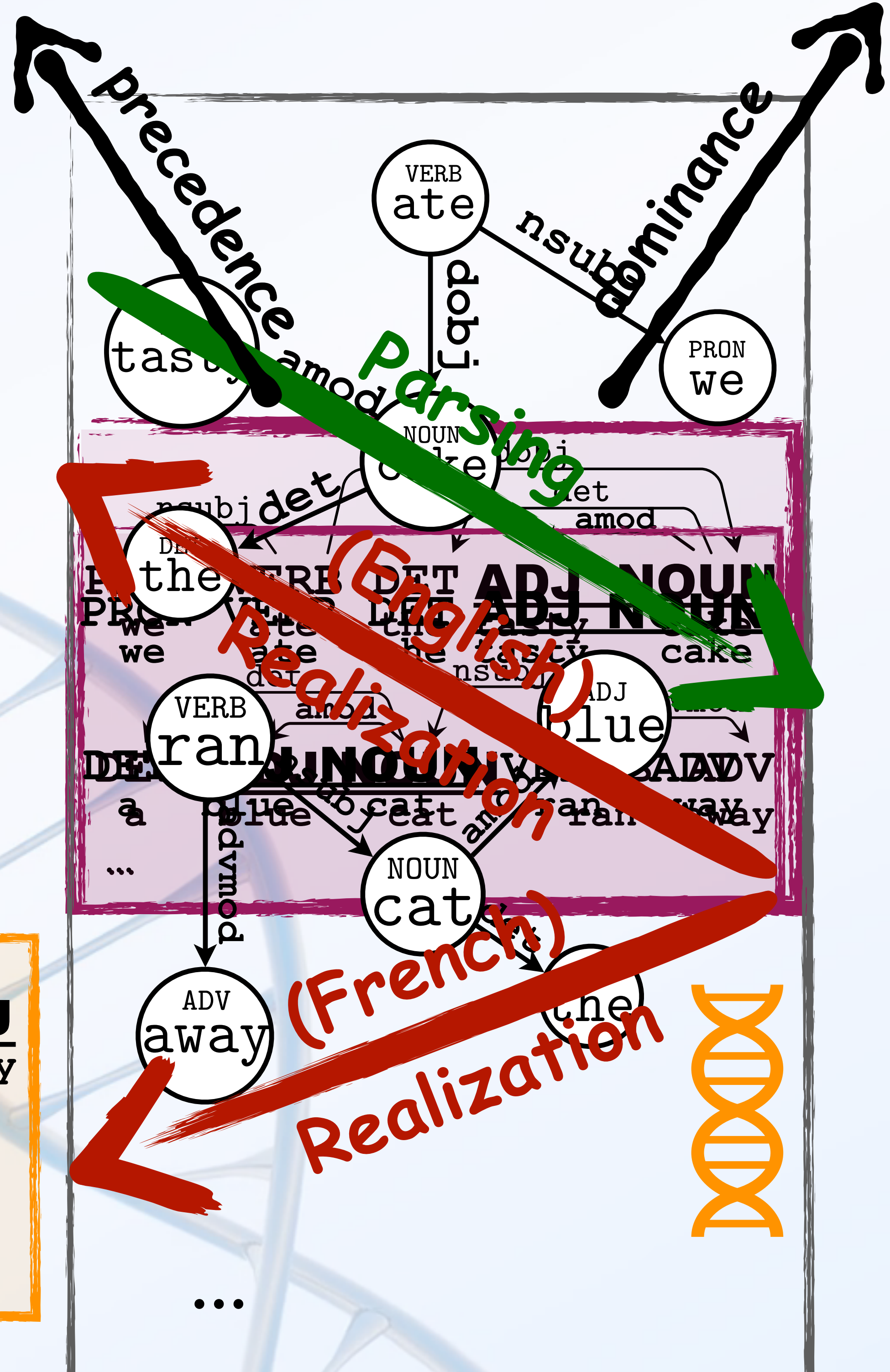
...

JOHNS HOPKINS
UNIVERSITY

# Synthetic Data

**Surface order**
(Lang. specific)

**Unordered dep.**
(Lang. universal)

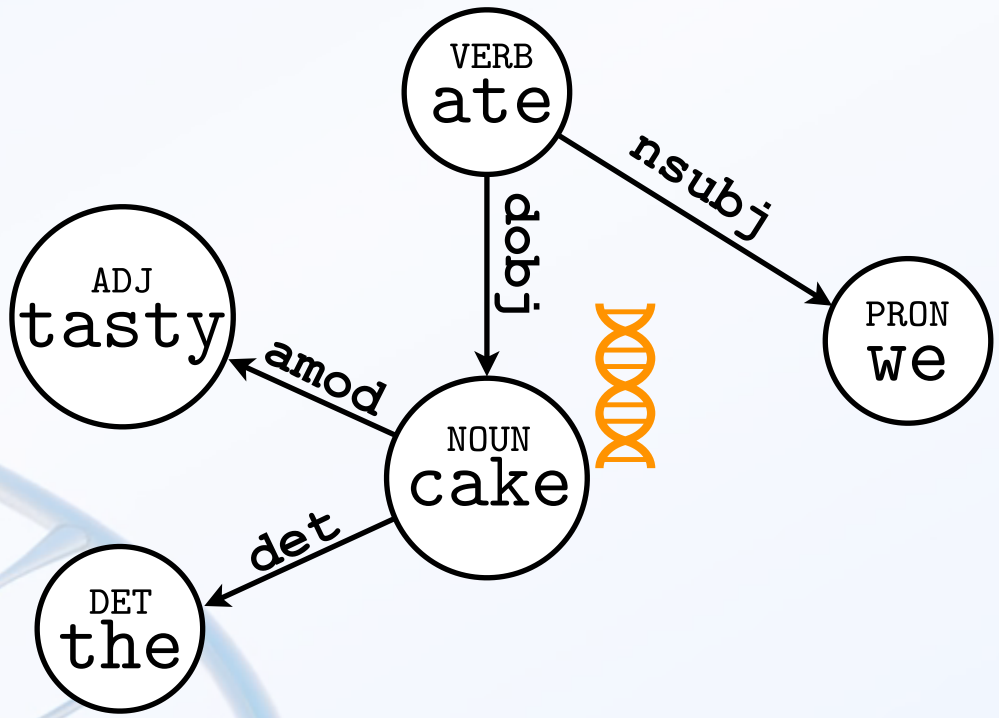



**PRON VERB DET NOUN ADJ**
we ate the cake tasty

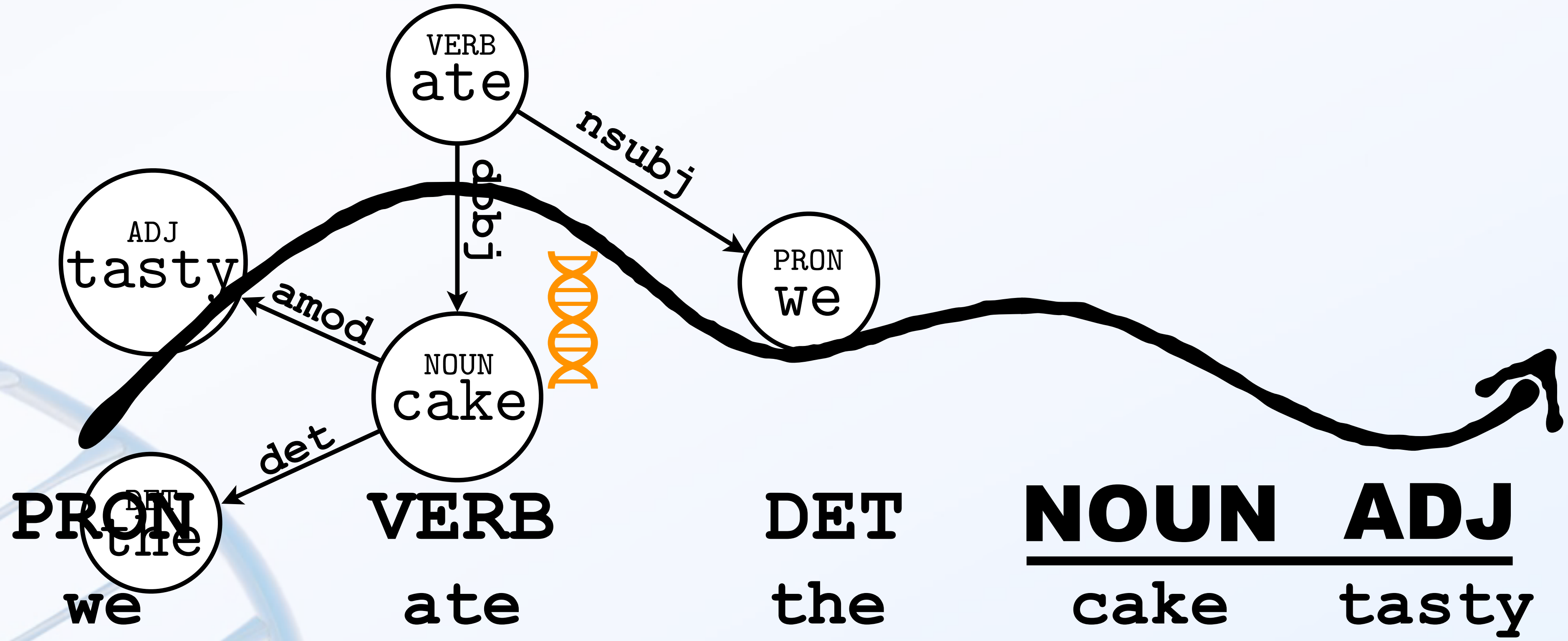
**DET NOUN ADJ VERB ADV**
a cat blue ran away

…

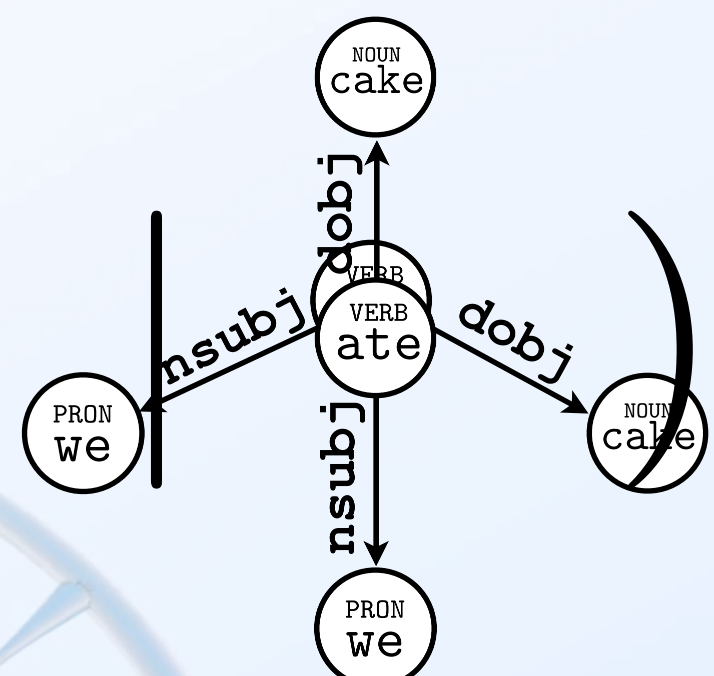
# Surface Realization



# Surface Realization




**Loglinear model**

$p\,($  $)$

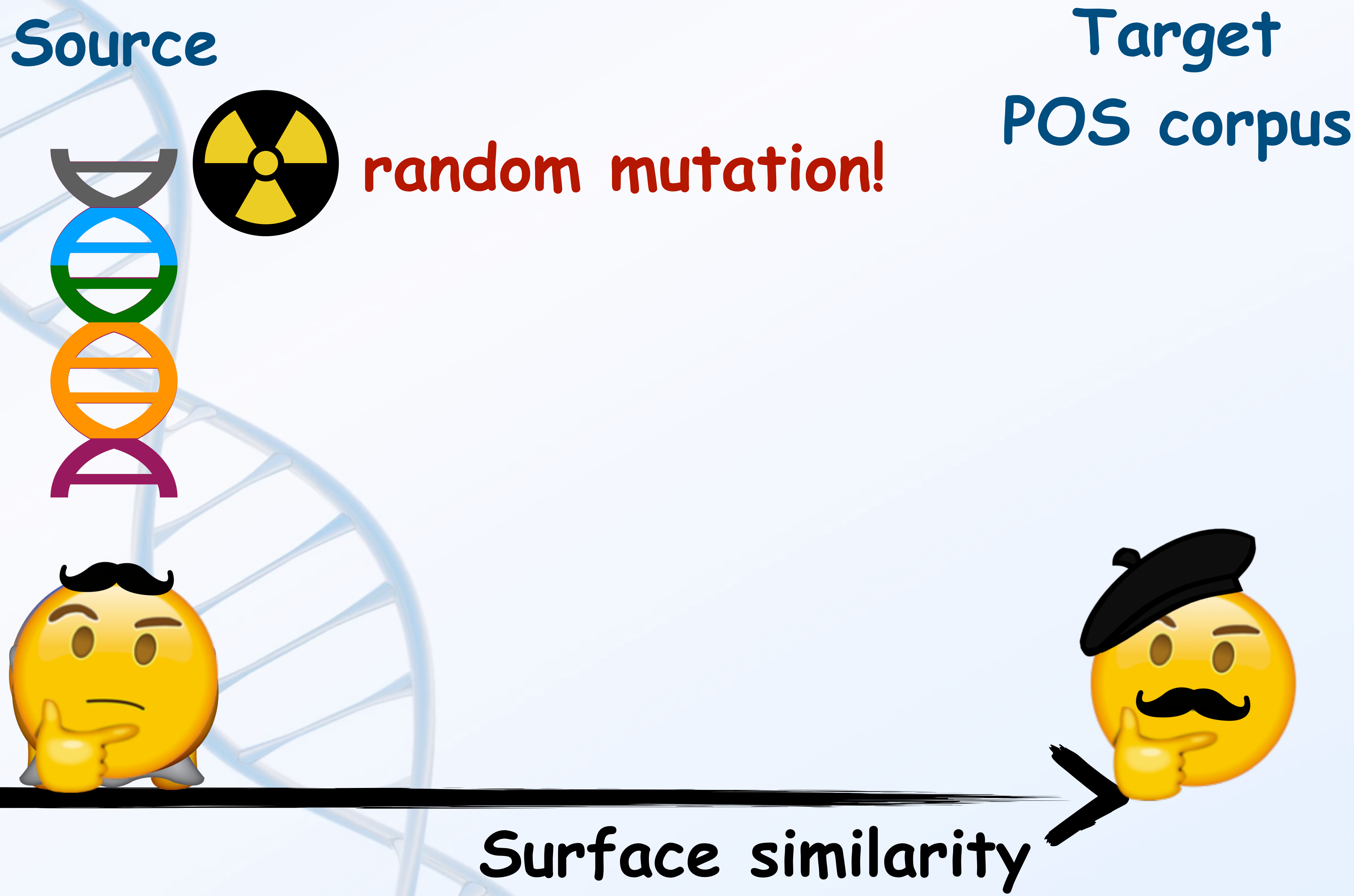
# How to find ?

Source'

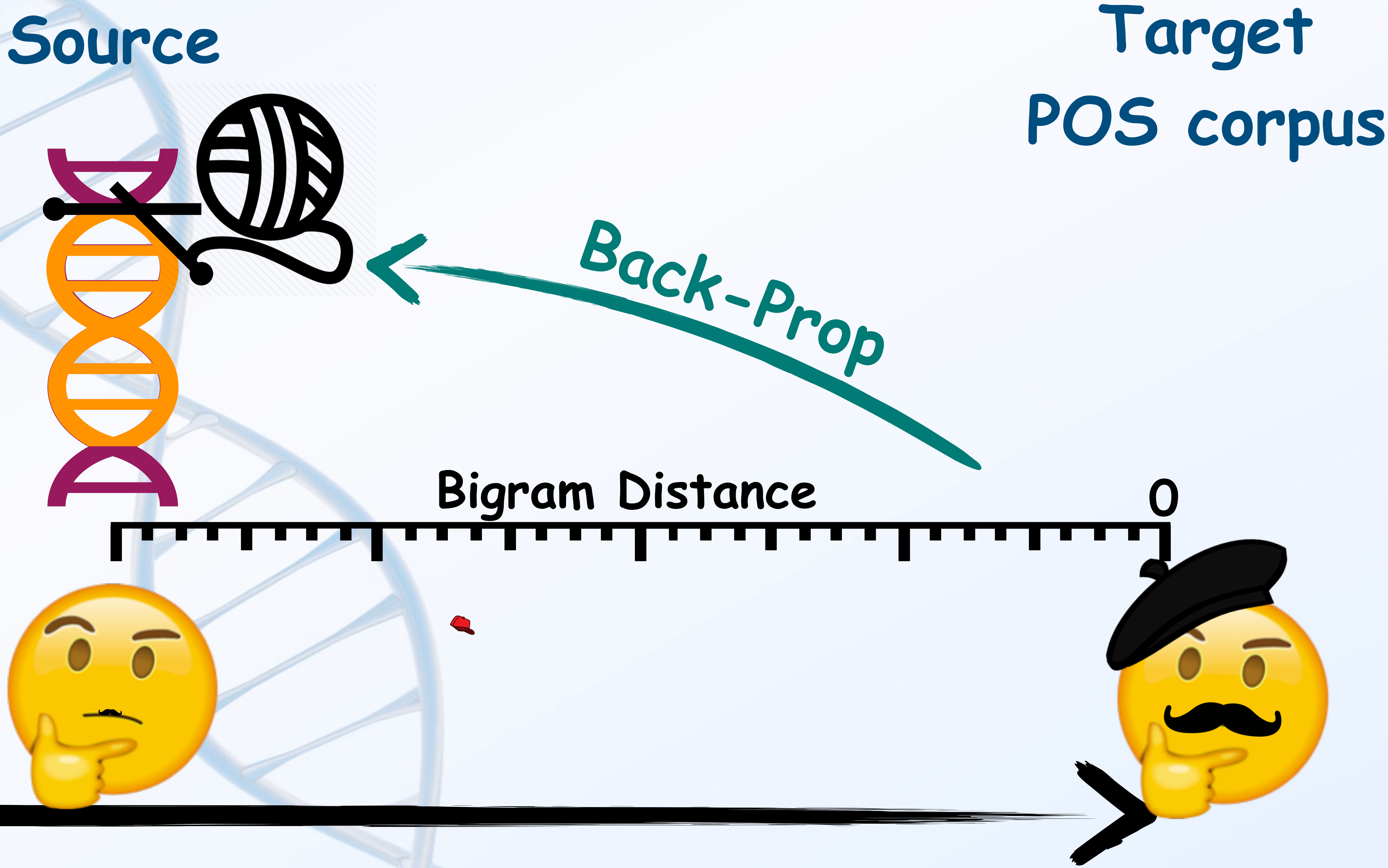
Target
POS corpus

Surface similarity

# Scattershot



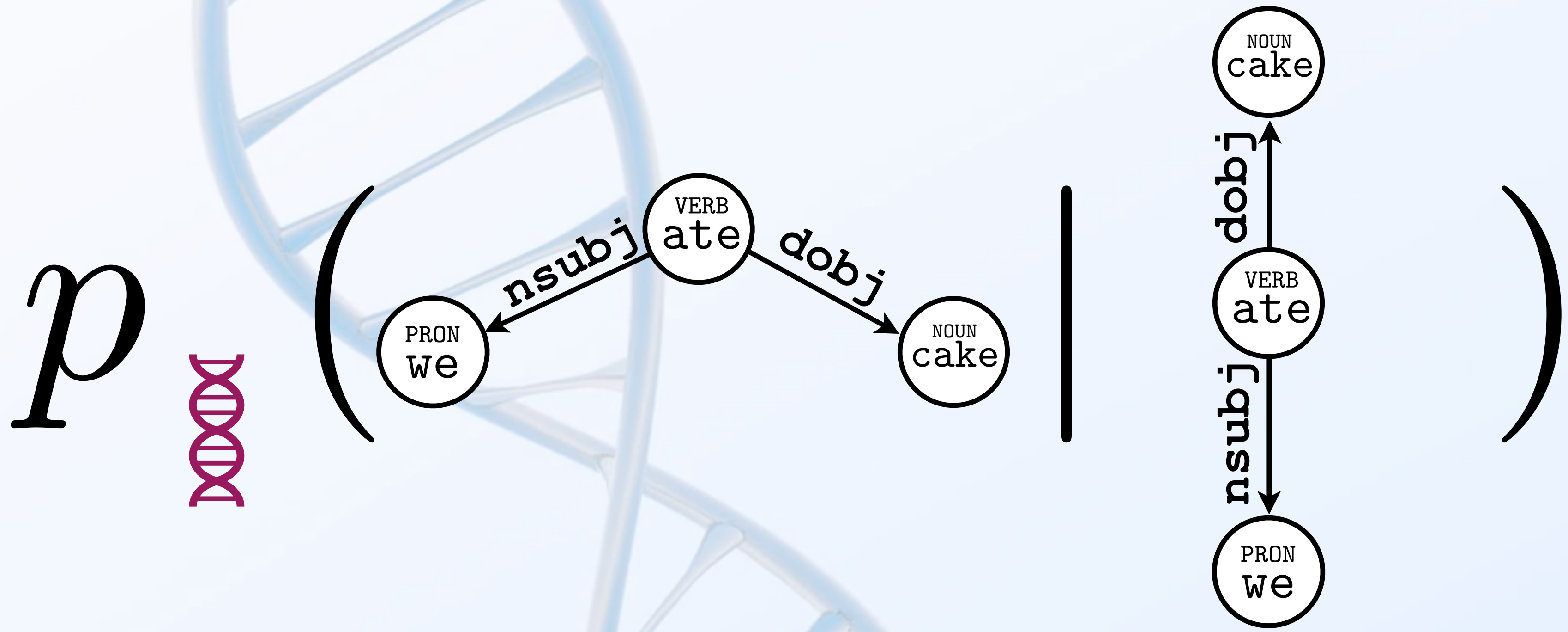
# This Paper: “Intelligent Design”



# Bigram Distance

- Whether the realized surface sentences have the similar POS-bigrams to the target language

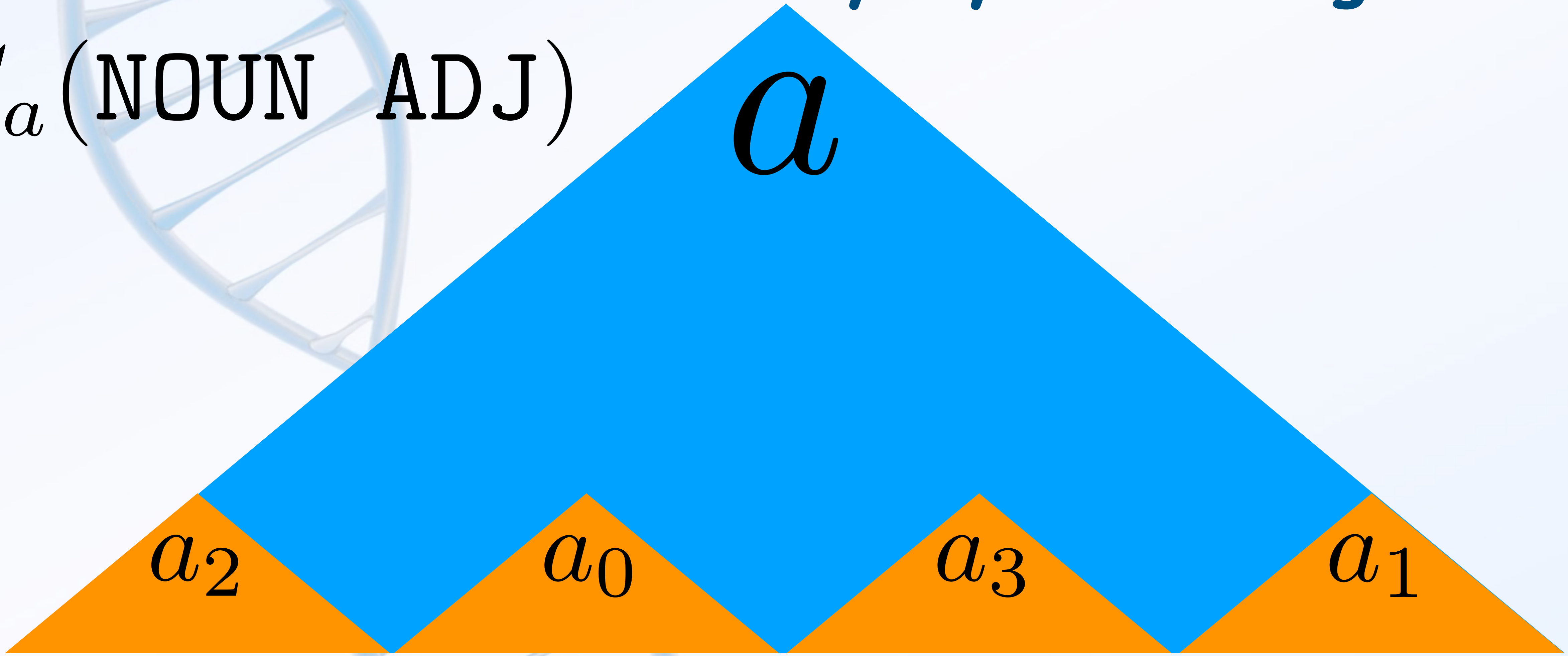
- How to compute the POS-bigrams counts?
  - **Expected Counts** from !



# Computing the Expected Counts

## by Dynamic Programming

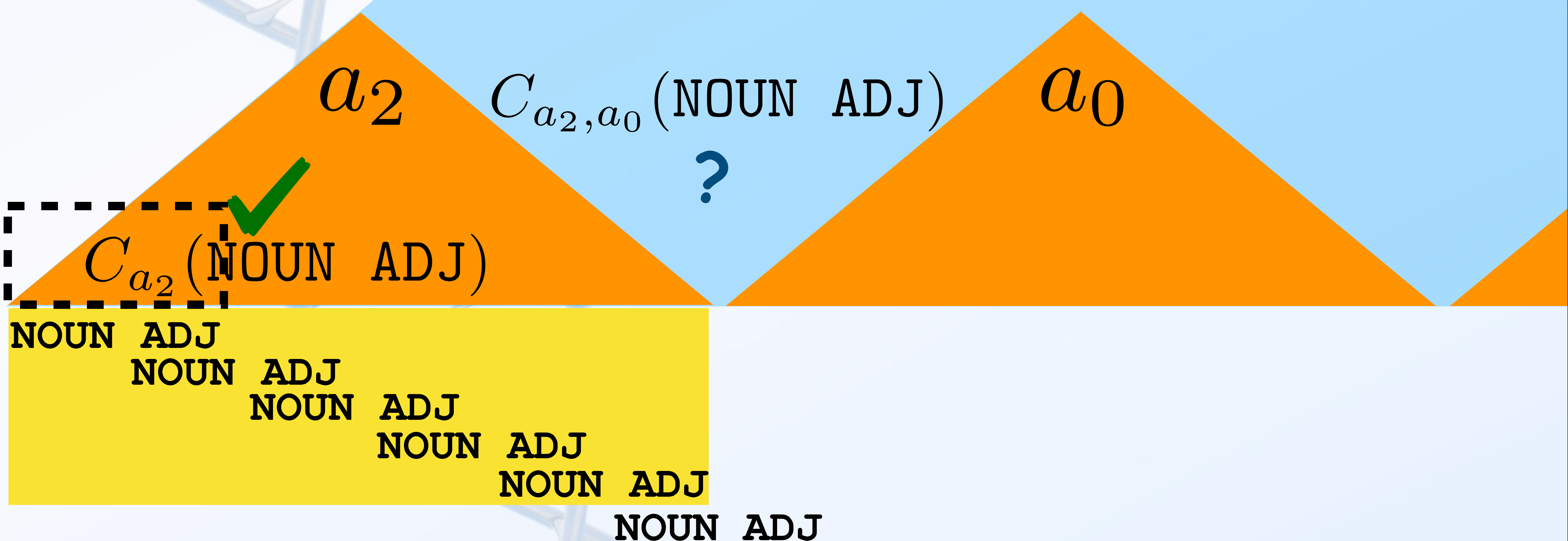
$C_a(\texttt{NOUN ADJ})$



# Computing the Expected Counts

## by Dynamic Programming

$$C_a(\texttt{NOUN ADJ})$$



# Computing the Expected Counts

## by Dynamic Programming

$C_{a_2,a_0}(\texttt{NOUN ADJ})$

$a_2$

$a_0$

$\texttt{\#}$ $\texttt{\#}$

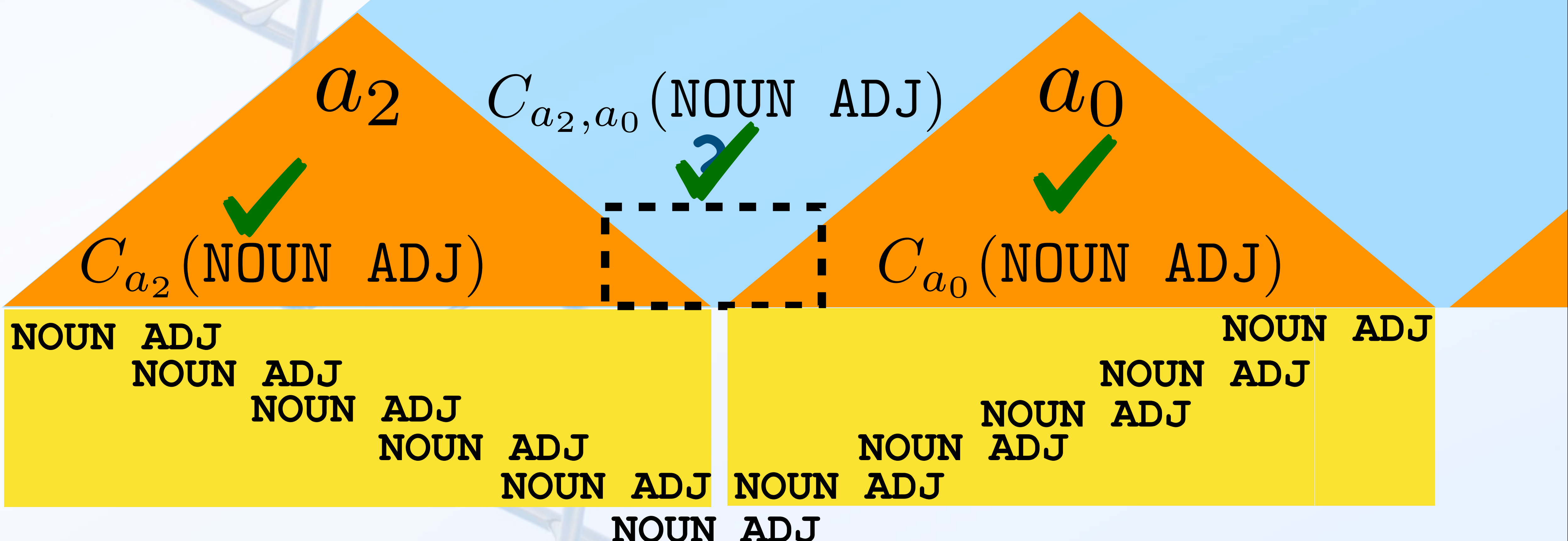
$$C_{a_2}(\texttt{NOUN \#}) \times C_{a_0}(\texttt{\# ADJ})$$

✔ ✔

**NOUN ADJ**

# Computing the Expected Counts
## by Dynamic Programming

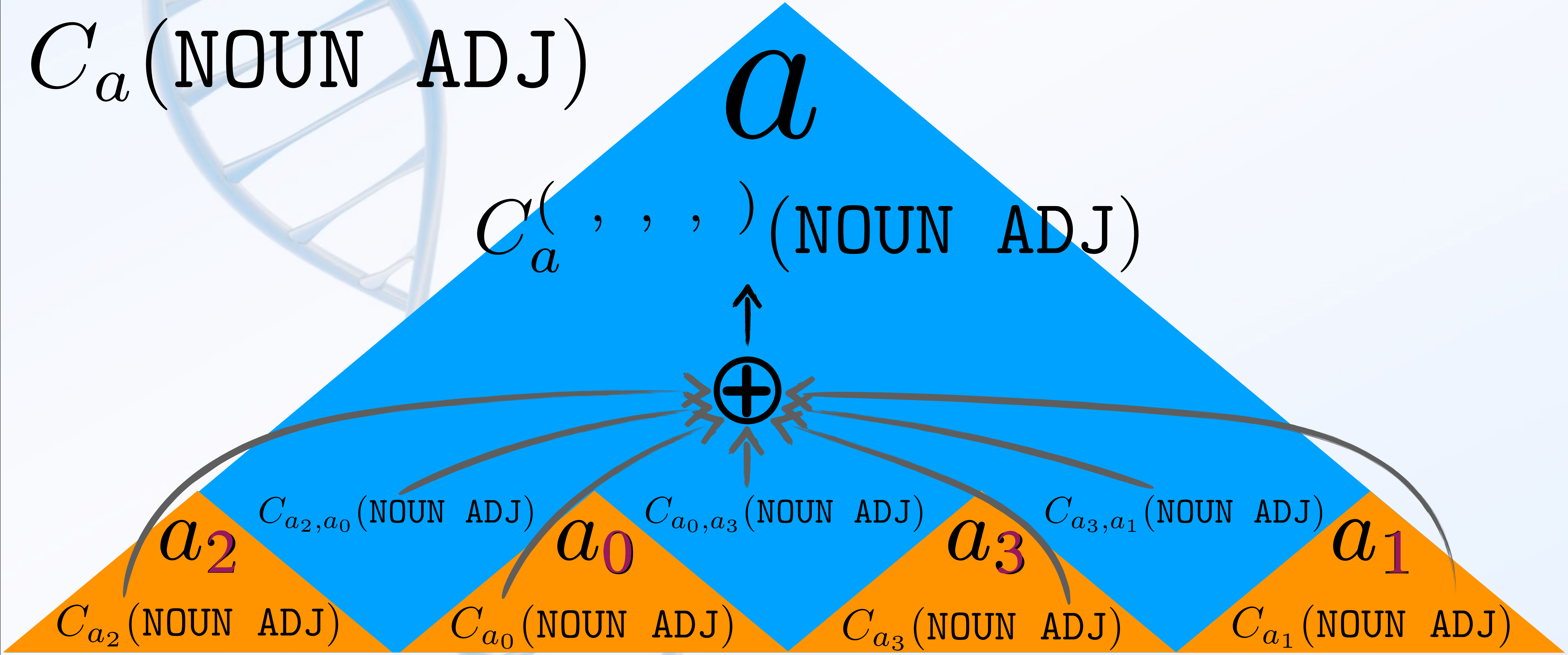
$$C_a(\texttt{NOUN ADJ})$$



# Computing the Expected Counts

## by Dynamic Programming

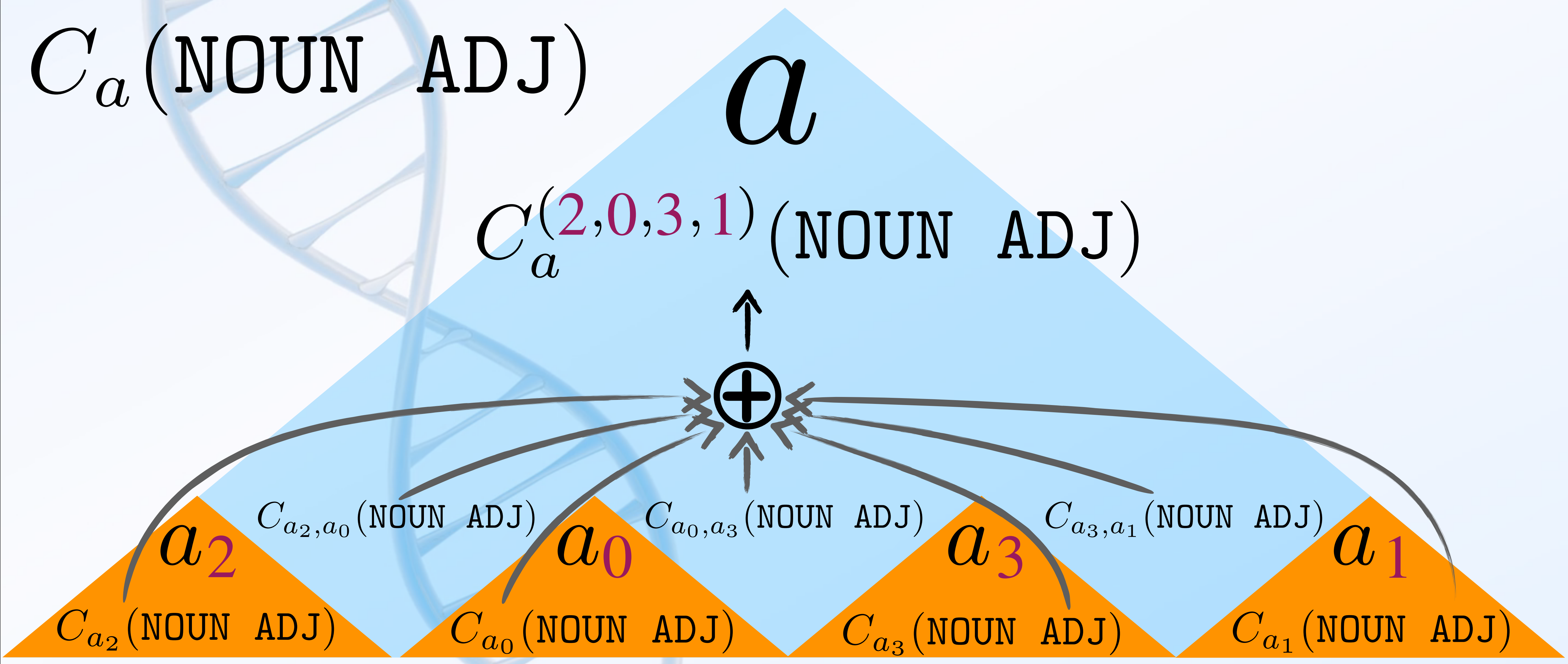
$C_a(\texttt{NOUN ADJ})$



# Computing the Expected Counts

## by Dynamic Programming

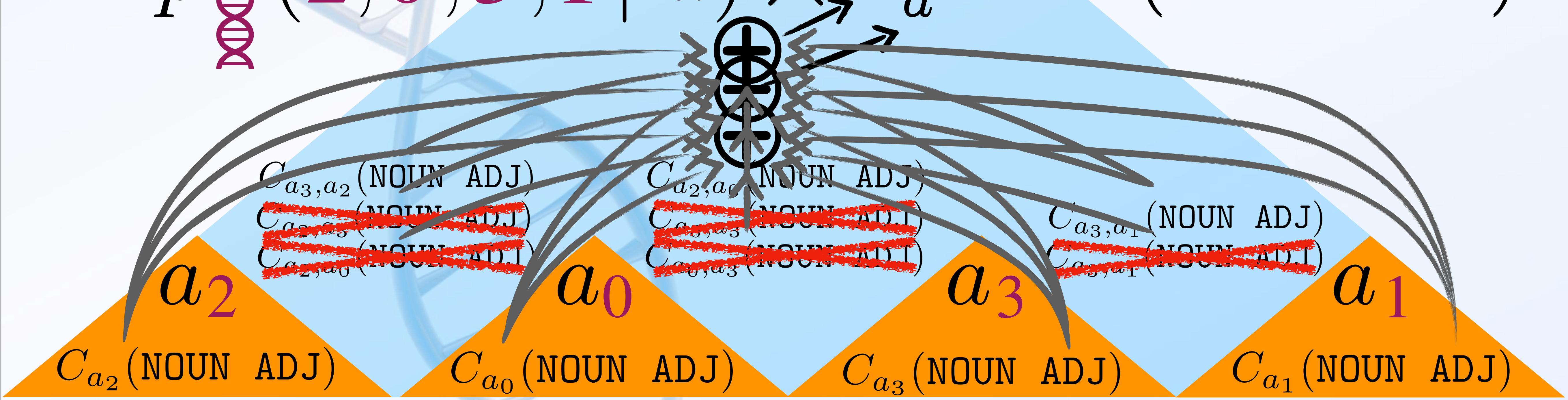
$C_a(\texttt{NOUN ADJ})$



# Computing the Expected Counts

## by Dynamic Programming

$$C_a(\texttt{NOUN ADJ}) \mathrel{+}= p(2,0,3,1 \mid a) \times C_a^{(2,0,3,1)}(\texttt{NOUN ADJ})$$



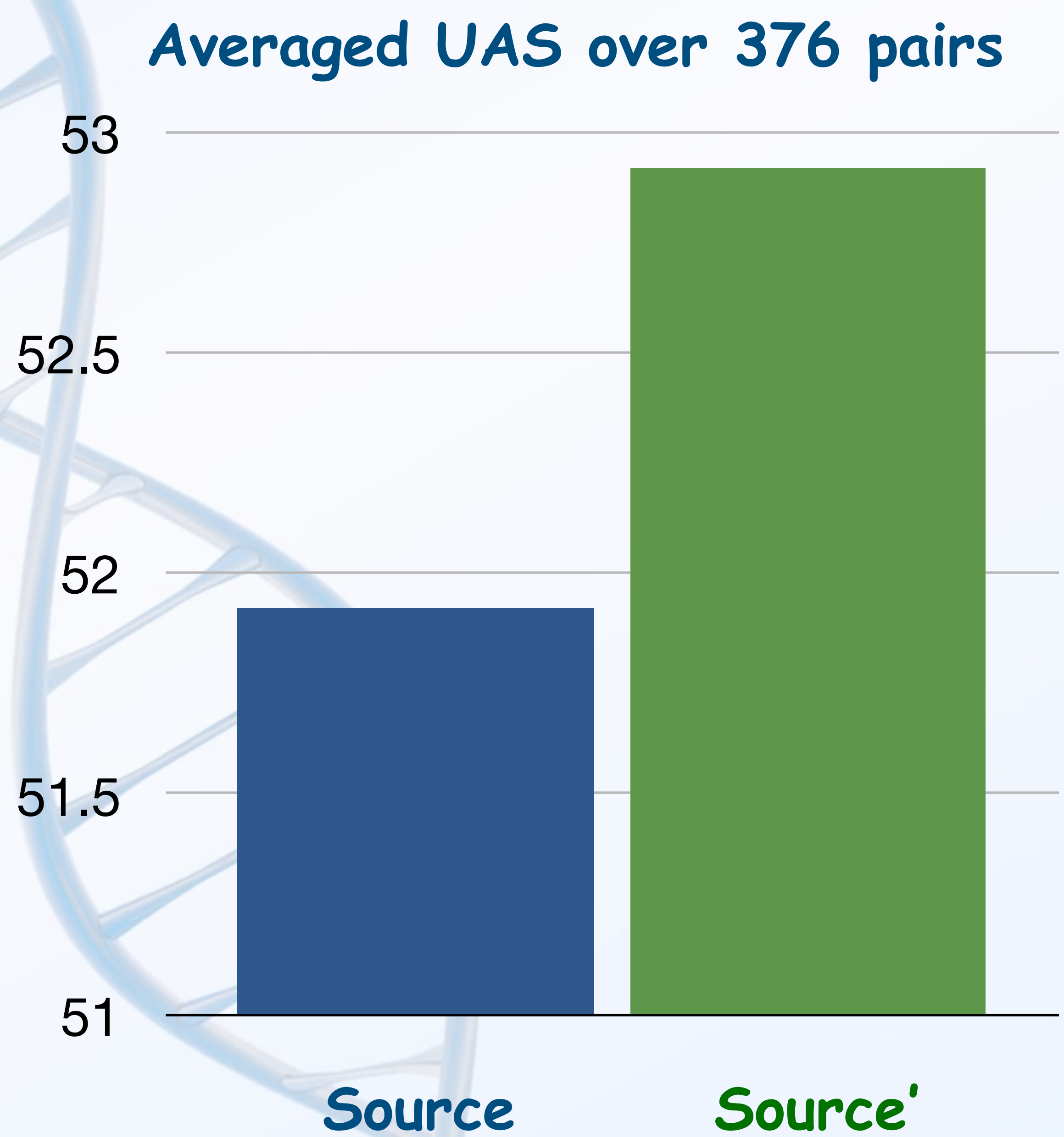

**4! Permutations**

# Data

- Universal Dependencies version 1.2
  - A collection of 37 dependency treebanks for 33 languages.

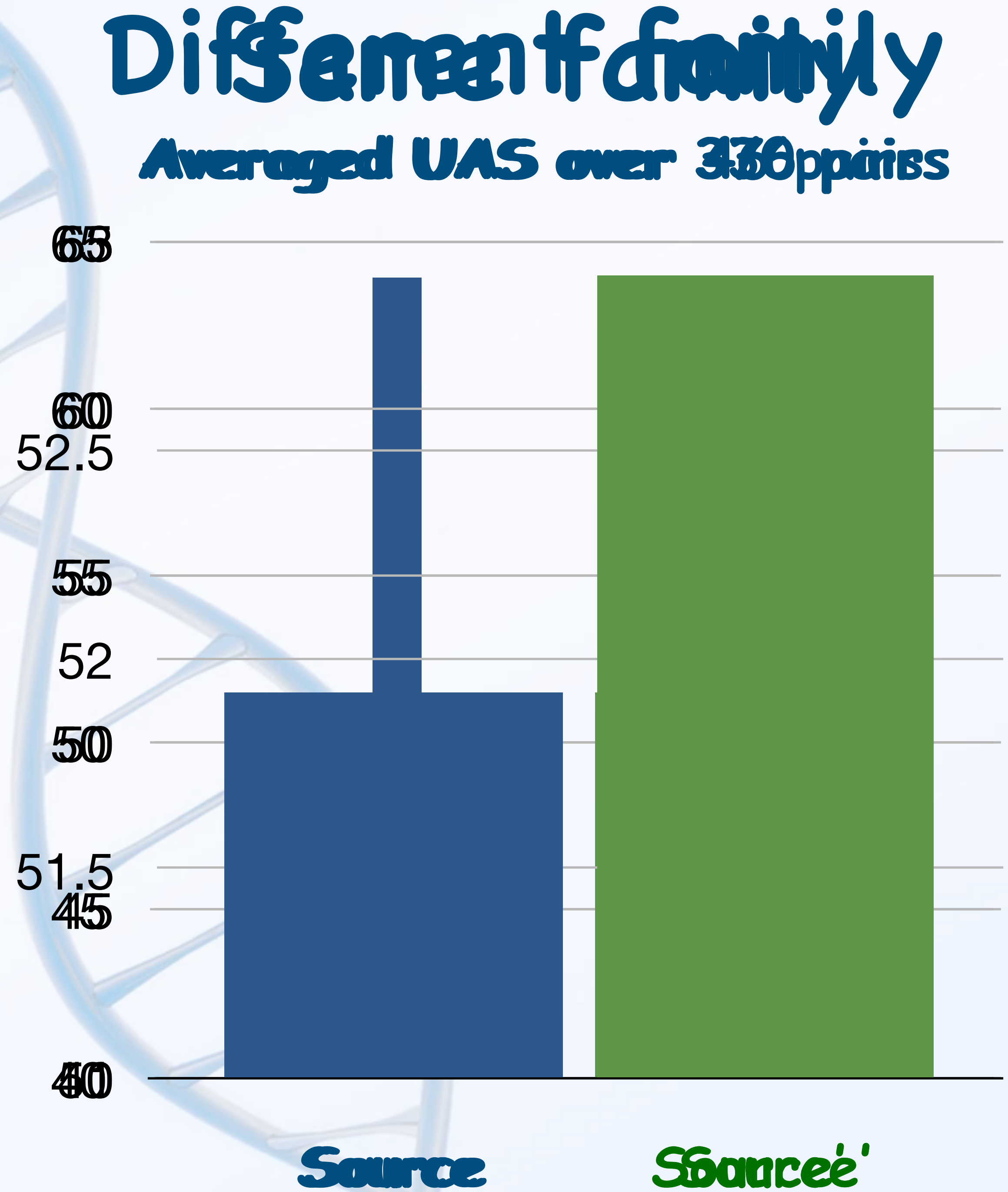
| Train | Test |
|---|---|
| cs, es, fr, hi, de, it, la_itt, no, ar, pt, en, nl, da, fi, got, grc, et, la_proiel, grc_proiel, bg | la, hr, ga, he, hu, fa, ta, cu, el, ro, sl, ja_ktc, sv, fi_ftb, id, eu, pl |

# Is your method work?

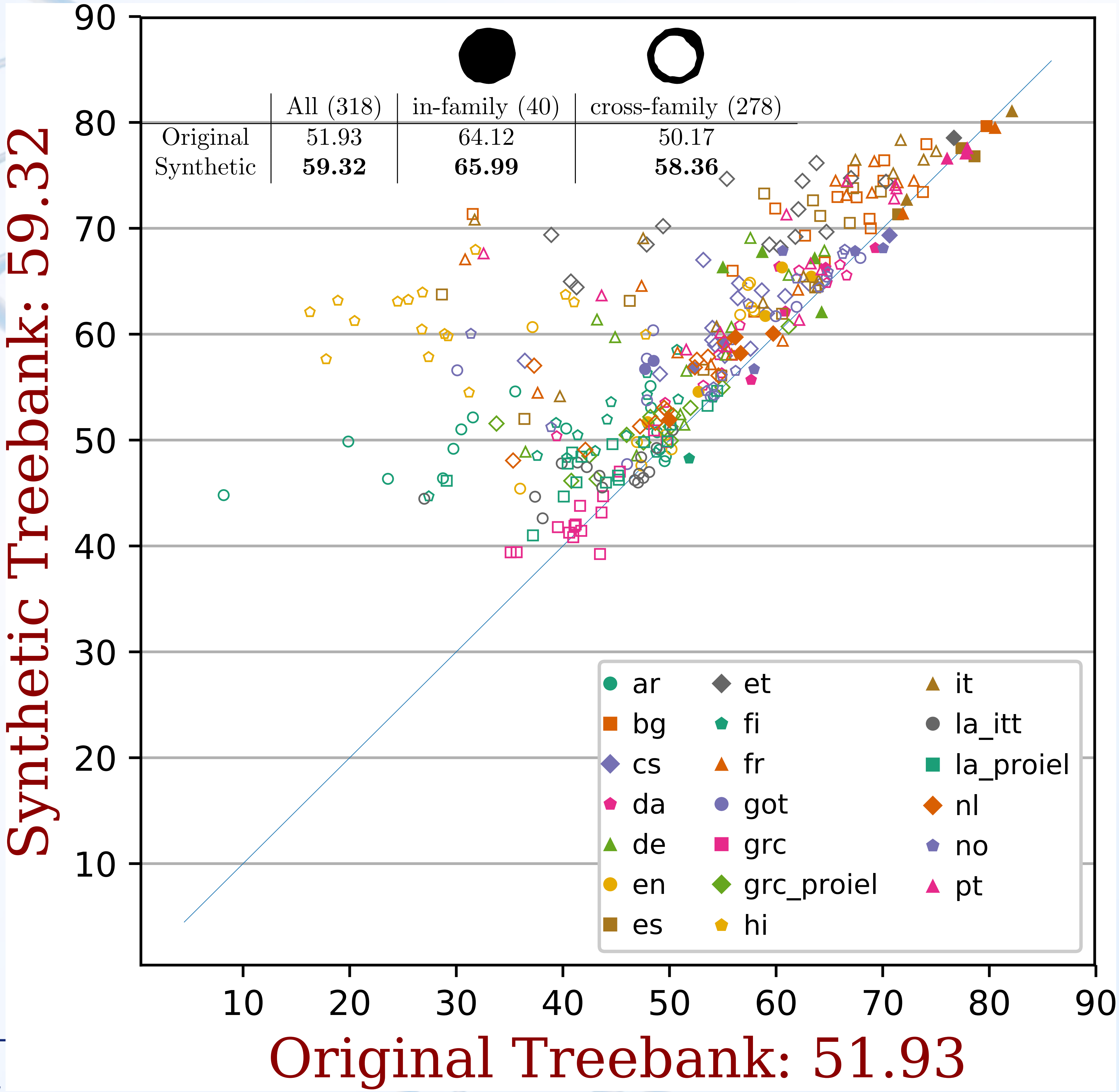



Overall YES!

# Depends on the language family



# Oracle Realization



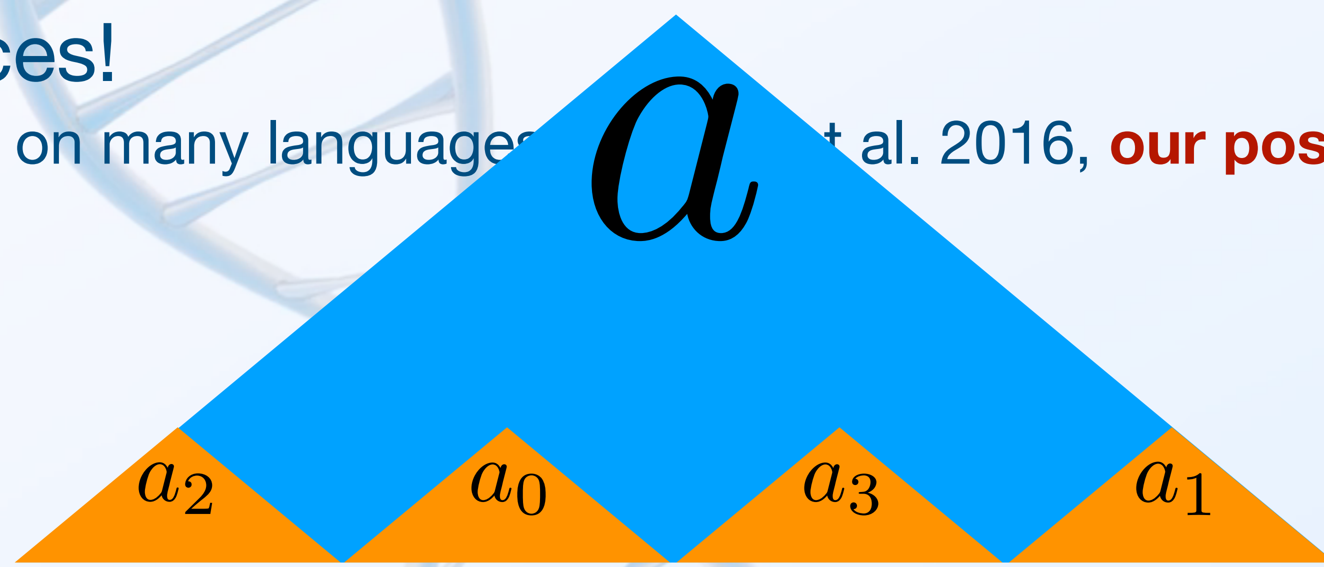
# Better parsing

- Fancier surface similarity function
  - Recurrent neural network language models capture longer context
  - Dynamic programming methods are no longer available.
  - We need approximate inference (sampling)!
- Relax the requirement of POS-tags
  - Unavailable for a low-resource target language
  - Richer lexical information would be better than POS-tags
  - Cross-lingual unsupervised word embeddings (Ruder et al., 2017) would be useful
- More efficient inference!
  - Enumerating over n! permutations
  - We could approximately sample from permutations (Eisner and Tromble, 2006)

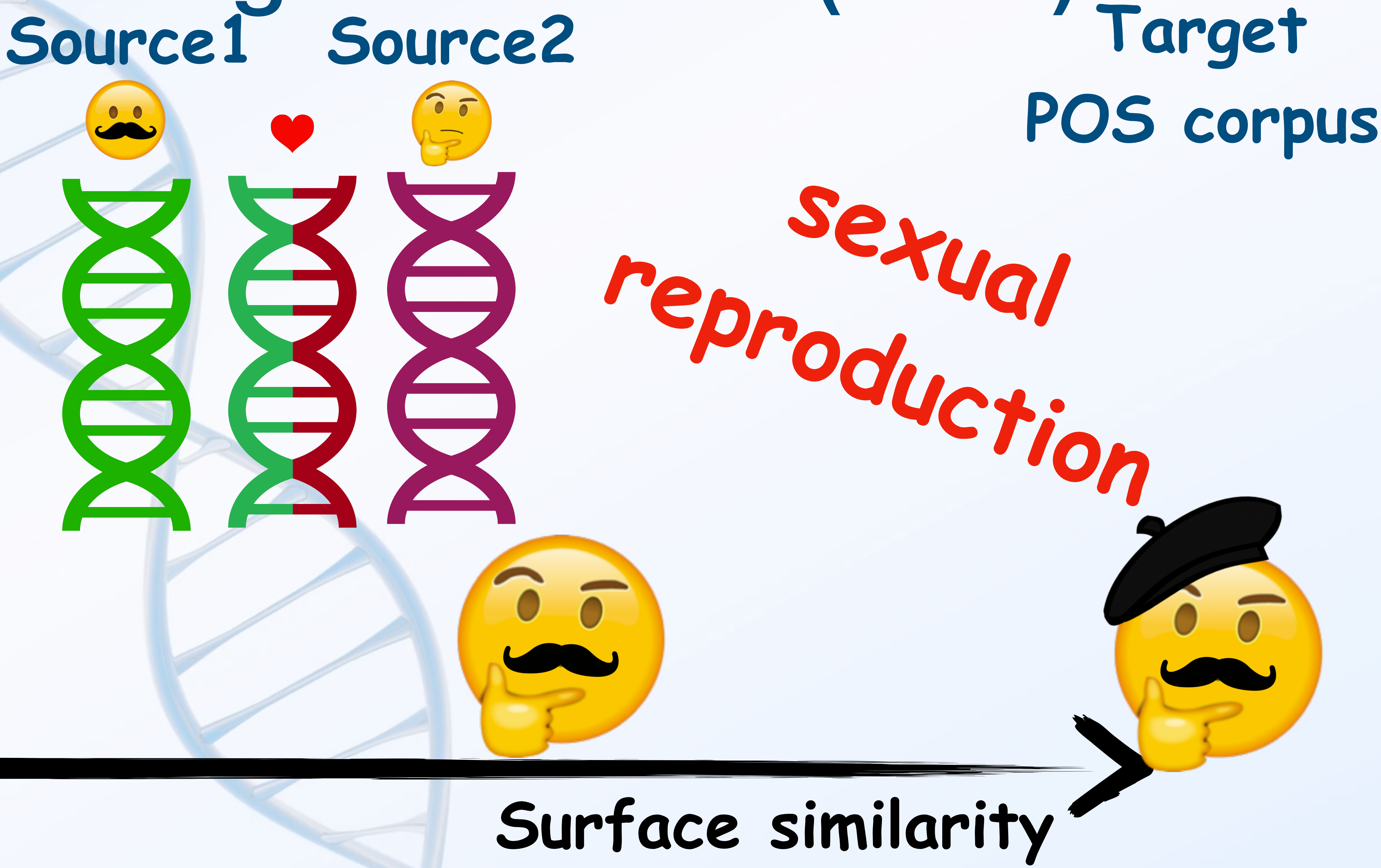
**NOUN VERB NOUN**

**SVO or OVS ?**

- Multi-sources!
  - One parser on many languages … al. 2016, **our poster tomorrow**)

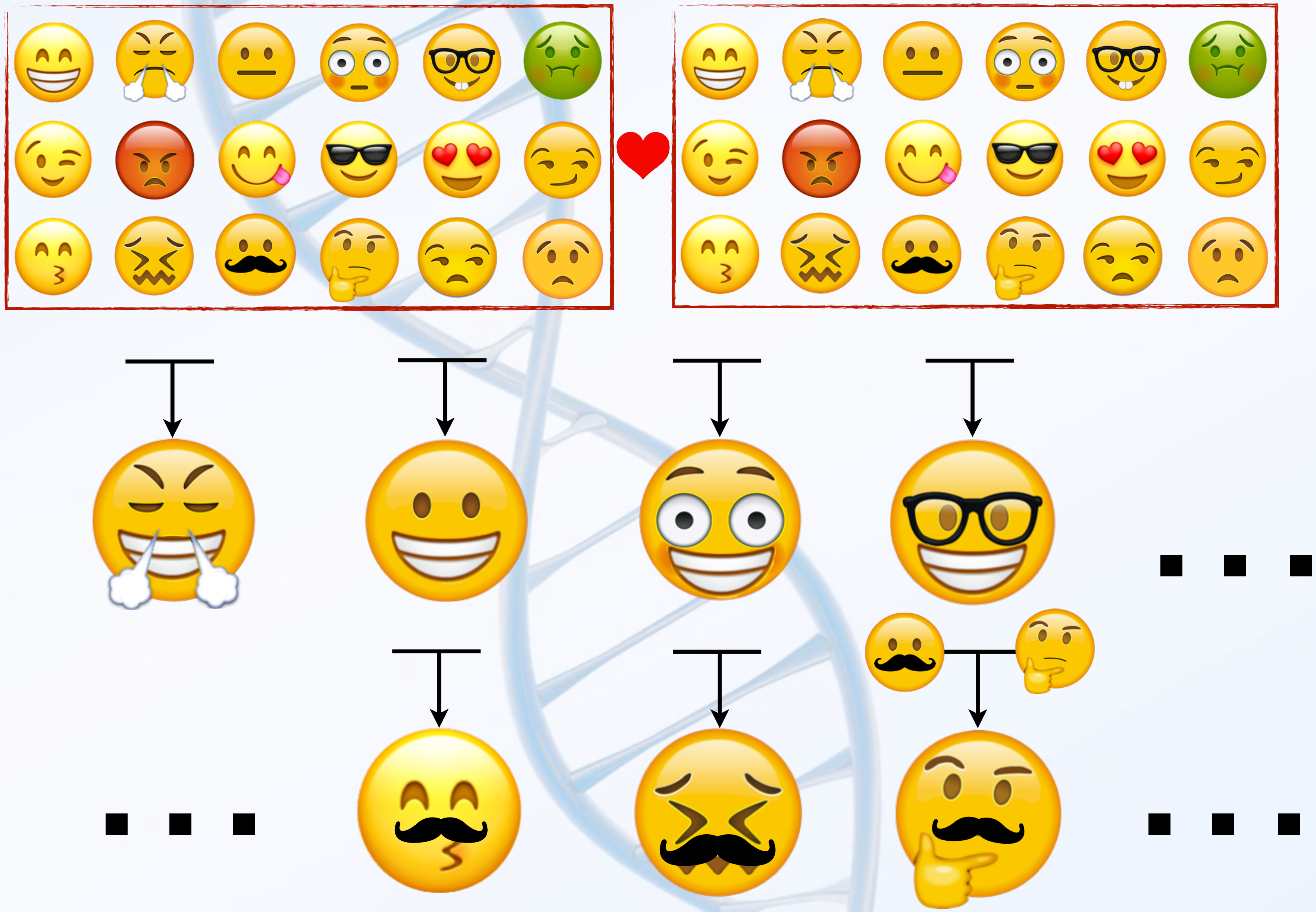


# Wang and Eisner (2016)



# Wang and Eisner (2016)

Source languages




Target
POS corpus

Wang and Eisner, **Surface statistics of an unknown language indicate how to parse it.** TACL 2018

# 11:00-12:30, Saturday, Grand Hall 2

# THANKS!

# Improving Cross-Linguistic Robustness by Training on Synthetic Languages

ACL “Typology for Polyglot NLP” workshop
August 2019


Jason Eisner

with

Dingquan Wang

JOHNS HOPKINS UNIVERSITY

CENTER FOR LANGUAGE AND SPEECH PROCESSING


TACL 2016, TACL 2017, TACL 2018, EMNLP 2018

# "What is a possible human language?"

- Typology in linguistics aims to <u>describe</u> the range of variation in human languages
  (Theories of Universal Grammar aim to <u>explain</u> it)

- If we know what languages <u>can</u> look like, we can make new ones!
- Training on realistic imagined languages is like dreaming. It can help us cope with the real languages we'll see when we wake up.

[switch here to EMNLP'18 slides]

# Surface similarity → deep similarity?

- By making English' like French superficially, we hope that a parser that finds correct English' trees will also find correct French trees.

## More generally: Surface properties as clues to deep properties

1. Surface typological feature: NOUN ADJ bigram is common
2. Deep typological feature: NOUN → ADJ modifier is common

- Is there an "implicational universal" linking 1 to 2?
- French has <u>many</u> surface features; by examining <u>many</u> languages, hope to learn what they imply about French

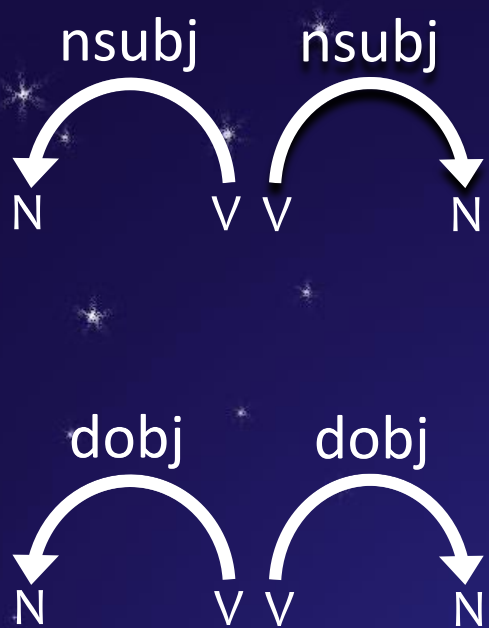
- Given sequences of part-of-speech (POS) tags, predict the basic word order of the language.

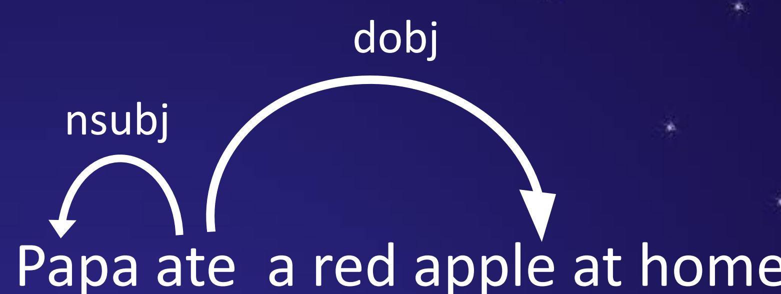
- What would you guess, based on your knowledge of how languages typically work?

Verb Det Noun Adj Det Noun

# Syntactic Typology (of English)

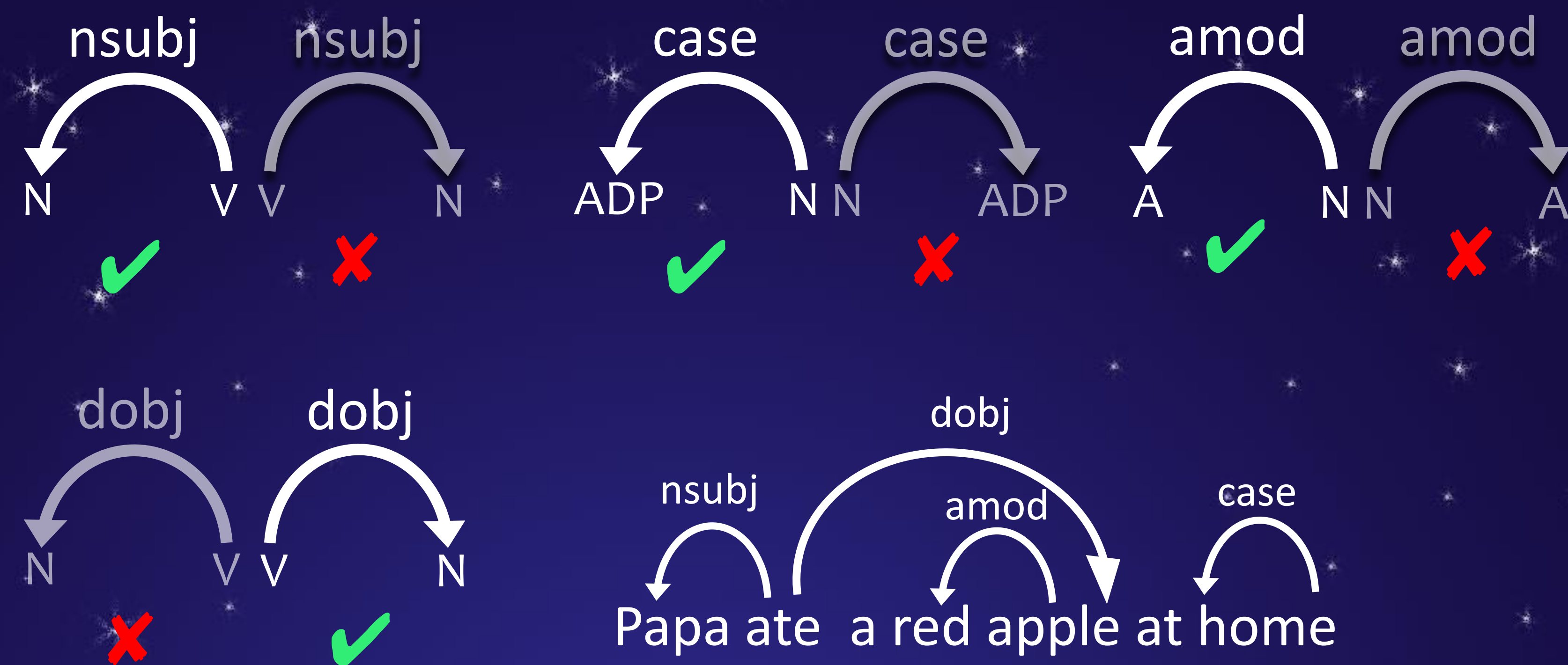
Subject-Verb-Object



# Syntactic Typology (of English)

Subject-Verb-Object | Prepositional | Adj-Noun



# Fine-grained Syntactic Typology (of English)

Subject-Verb-Object

Prepositional

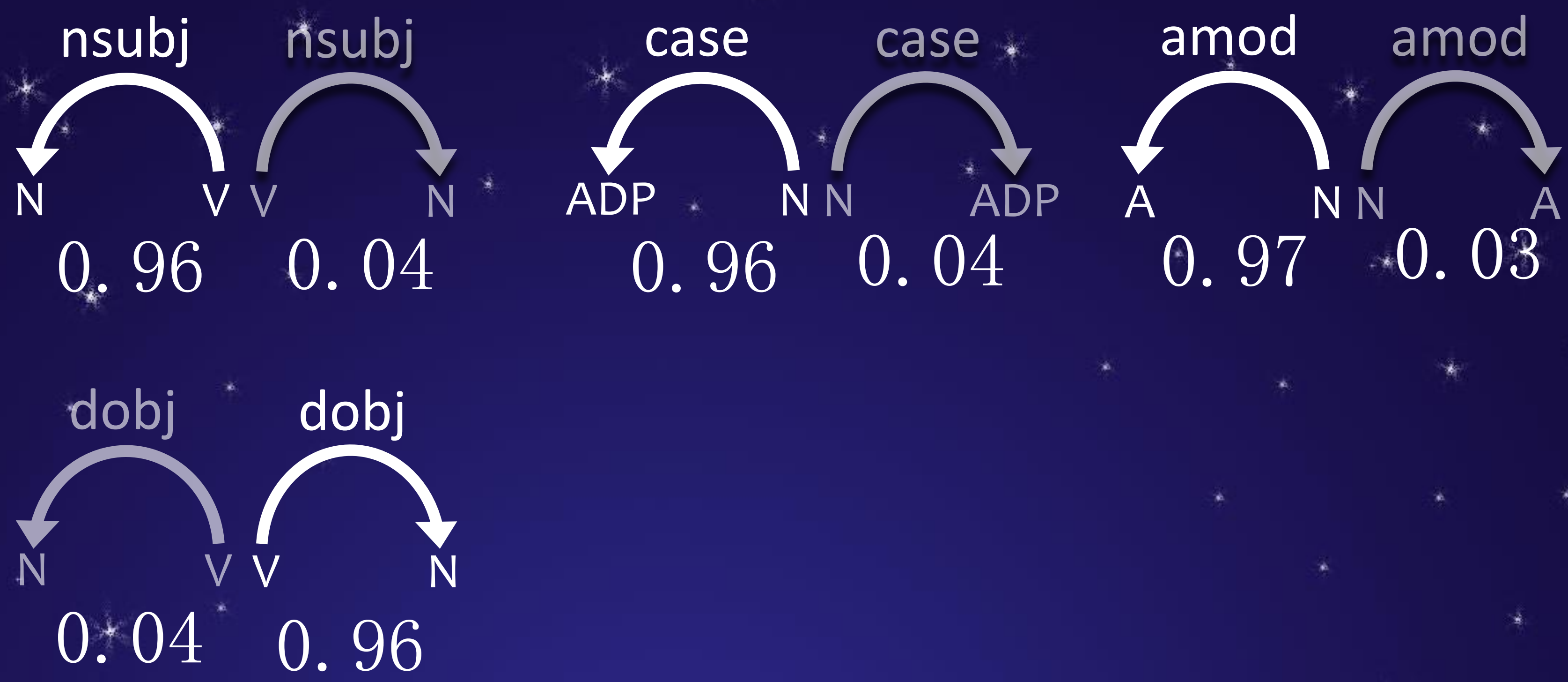
Adj-Noun

# Fine-grained Syntactic Typology (of English)

Subject-Verb-Object

Prepositional

Adj-Noun



# Fine-grained Syntactic Typology (of English)

Subject-Verb-Object

Prepositional

Adj-Noun





Vector of length 57

| nsubj | dobj | case | amod | ... |
|---|---|---|---|---|
| 0.04 | 0.96 | 0.04 | 0.03 | ... |

# Fine-grained Syntactic Typology (of Japanese)

Subject-Object-Verb

nsubj

V N

0. 0

dobj

V N

0. 0

Postpositional

case

N ADP

1. 0

Adj-Noun

amod

N A

0. 0

Vector of length 57

| nsubj | dobj | case | amod | ... |
|---|---|---|---|---|
| 0.0 | 0.0 | 1.0 | 0.0 | ... |

# Fine-grained Syntactic Typology (of Hindi)

Subject-Object-Verb

Postpositional

Adj-Noun

nsubj

V N

0. 01

case

N ADP

0. 98

amod

N A

0. 03

Vector of length 57

dobj

V N

0. 25

| nsubj | dobj | case | amod | … |
|---|---|---|---|---|
| 0.01 | 0.25 | 0.98 | 0.03 | … |

# Fine-grained Syntactic Typology (of French)

Subject-Verb-Object | Prepositional | Noun-Adj




Vector of length 57


| nsubj | dobj | case | amod | … |
|---|---|---|---|---|
| 0.03 | 0.76 | 0.01 | 0.73 | … |

# Fine-grained Syntactic Typology

| Language | Typology | | | | |
|---|---|---|---|---|---|
| | nsubj | dobj | case | amod | ... |
| English | 0.04 | 0.96 | 0.04 | 0.03 | ... |
| Japanese | 0.0 | 0.0 | 1.0 | 0.0 | ... |
| Hindi | 0.01 | 0.25 | 0.98 | 0.03 | ... |
| French | 0.03 | 0.76 | 0.01 | 0.73 | ... |

# Fine-grained Syntactic Typology

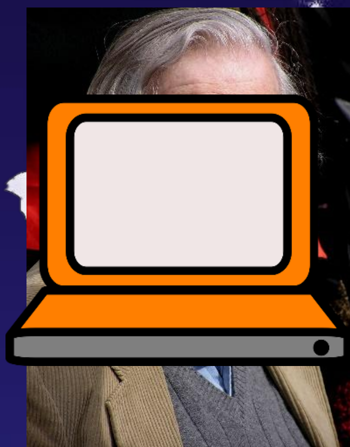
## Corpus of tags: ũ — Typology

| Corpus of tags | nsubj | dobj | case | amod | … |
|---|---|---|---|---|---|
| • NOUN VERB ADP NOUN PUNCT<br>• NOUN VERB PART NOUN PUNCT<br>… | 0.04 | 0.96 | 0.04 | 0.03 | … |
| • NOUN DET NOUN VERB PUNCT<br>• NOUN NOUN VERB PART<br>… | 0.0 | 0.0 | 1.0 | 0.0 | … |
| • NOUN AUX NOUN ADP PUNCT<br>• AUX NOUN NUM NOUN VERB<br>… | 0.01 | 0.25 | 0.98 | 0.03 | … |
| • NOUN VERB ADP NOUN PUNCT<br>• NOUN VERB NOUN PUNCT<br>… | 0.03 | 0.76 | 0.01 | 0.73 | … |

# Traditional approach: Grammar induction

- Yer/PRON amos/AUX yjja/VERB Ajjx/PROPN aat/ADP orrr/PRON ./PUNCT
- Per/NOUN anni/VERB inn/ADP se/NOUN in/PART hahh/CASE wee/VERB ./PUNCT
- Con/VERB per/NOUN aat/ADP Ajjx/PROPN “/PUNCT tat/PRON “/PUNCT yue/ADP han/NOUN ./PUNCT

…



SVO

| | |
|---|---|
| S → NP VP | 0.9 |
| VP → VP PP | 0.9 |
| … | |

•

•

•

JOHNS HOPKINS UNIVERSITY

# Grammar Induction

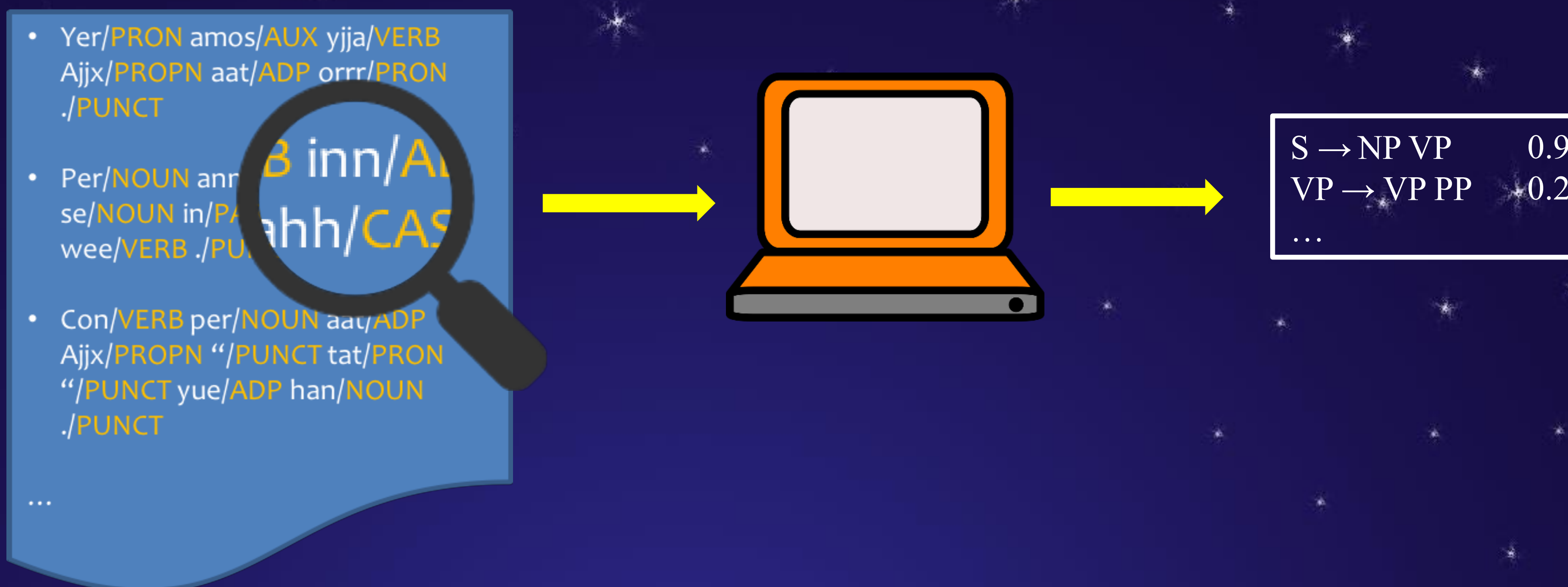
- Yer/PRON amos/AUX yjja/VERB Ajjx/PROPN aat/ADP orrr/PRON ./PUNCT
- Per/NOUN anni/VERB inn/ADP se/NOUN in/PART hahh/CASE wee/VERB ./PUNCT
- Con/VERB per/NOUN aat/ADP Ajjx/PROPN “/PUNCT tat/PRON “/PUNCT yue/ADP han/NOUN ./PUNCT

…

| | |
|---|---|
| S → NP VP | 0.9 |
| VP → VP PP | 0.2 |
| … | |

# Grammar Induction

- Unsupervised method (like EM)



# Grammar Induction

- Unsupervised method (like EM)
  - Converges on hypothesized trees
  - Just read the word order off the trees!
  - Alas, works terribly!

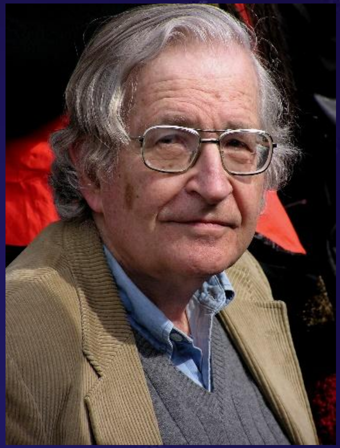
- Why doesn’t grammar induction work (yet)?
  - Locally optimal
  - Hard to harness linguistic knowledge
    - Doesn’t use any evidence outside the corpus
  - Might use the latent variables in the “wrong” way
    - Won't follow syntactic conventions used by linguists
    - Might not even model syntax, but other things like topic

# So how were <u>you</u> able to do it?

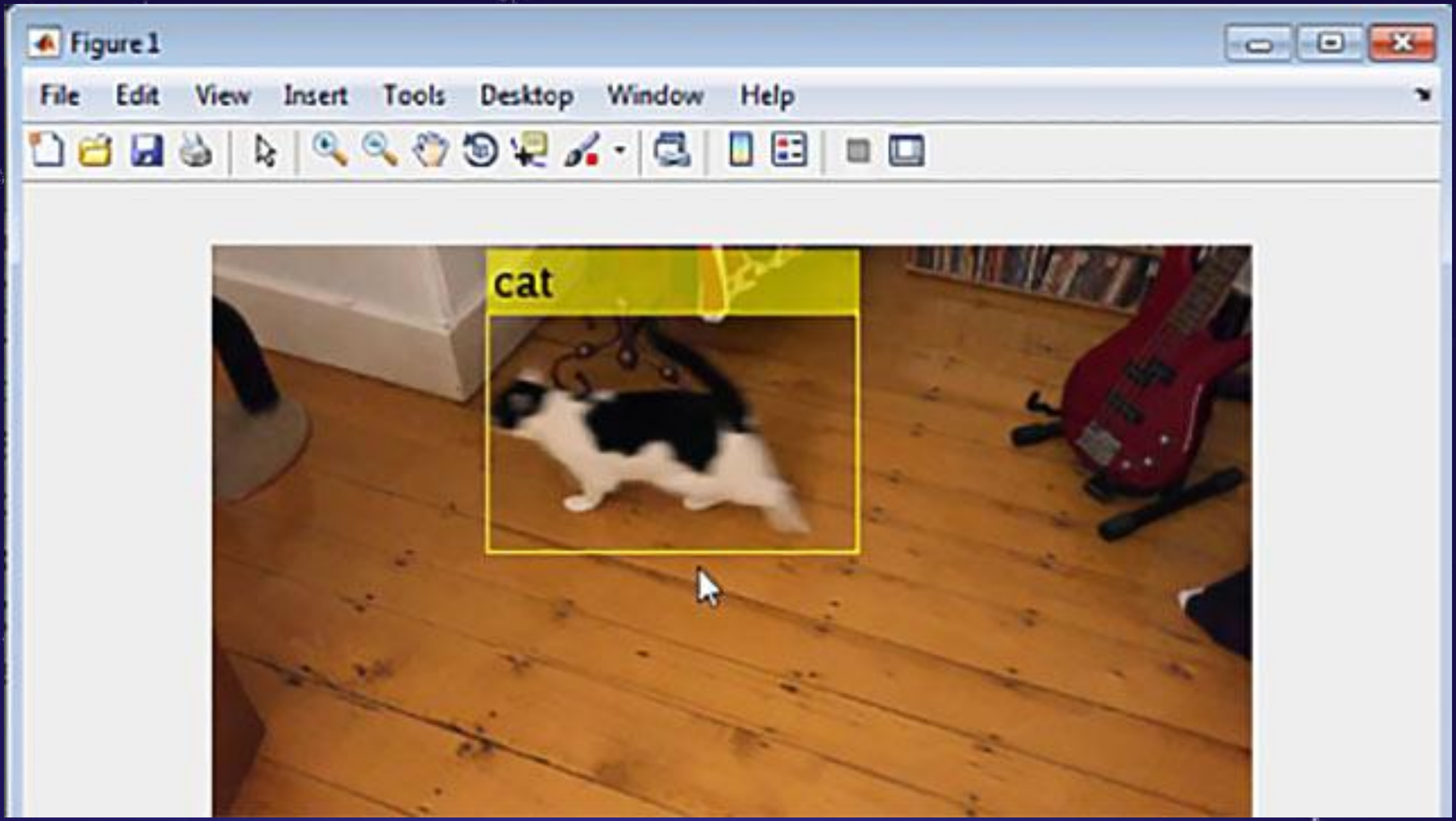
- It seems like linguists might be able:

Verb Det Noun Adj Det Noun

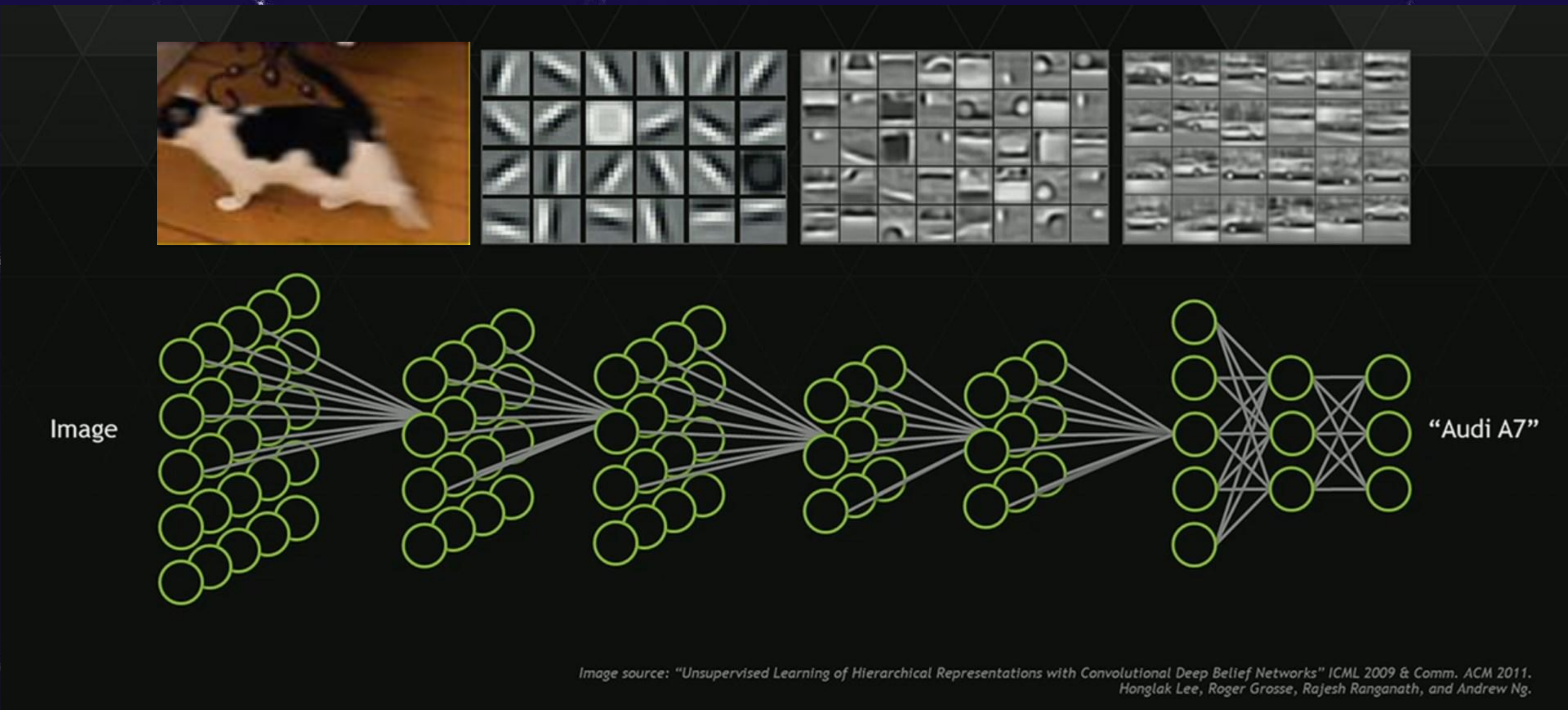
- Verb at start of sentence
- Noun-Adj bigram; Adj-Det bigram
- **Are simple cues like this useful?**
  - Principles & Parameters (1981)
  - Triggers (1994, 1996, 1998)



# Not holding out hope for a single trigger



# But a combination of cues might work



# Supervised learning

- Yer/PRON amos/AUX yjja/VERB Ajjx/PROPN aat/ADP orrr/PRON ./PUNCT
- Per/NOUN ann… se/NOUN in/PA… wee/VERB ./PU…
- Con/VERB per/NOUN aat/ADP Ajjx/PROPN “/PUNCT tat/PRON “/PUNCT yue/ADP han/NOUN ./PUNCT

…

…B inn/A…
…ahh/CAS…

| 0.25 | 0.8 | 1.0 | … |
|---|---|---|---|

training data

( 
- You/PRON can/AUX …
- Keep/VERB Google/PROPN …
- In/ADP my/PRON office/NOUN …

… ,

| 0.04 | 0.96 | 0.04 | … |
|---|---|---|---|

)

( 
- /PRON /AUX …
- /VERB /PROPN
- /ADP /PRON /NOUN …

… ,

| 0.03 | 0.76 | 0.01 | … |
|---|---|---|---|

)

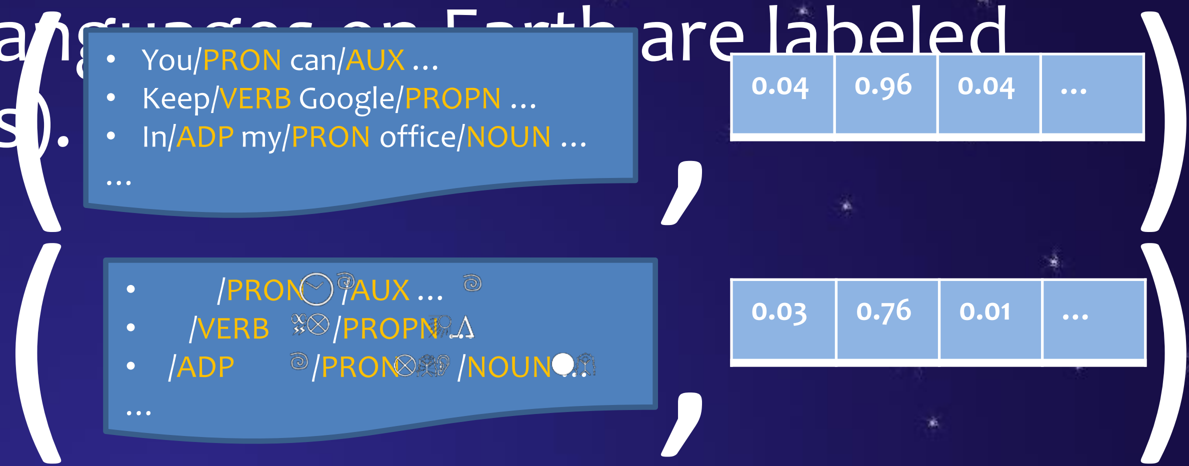
# From Unsupervised to Supervised

- Unsupervised method (like EM)
  - Locally optimal
  - Hard to harness linguistic knowledge
  - Might use the latent variables in the "wrong" way
    - Won't follow syntactic conventions used by linguists
    - Might not even model syntax, but other things like topic
- How about a supervised method?

  - Globally optimal (if objective is convex)
  - Allows feature-rich discriminative model
  - Imitates what it sees in **supervised training data**

# What's wrong?

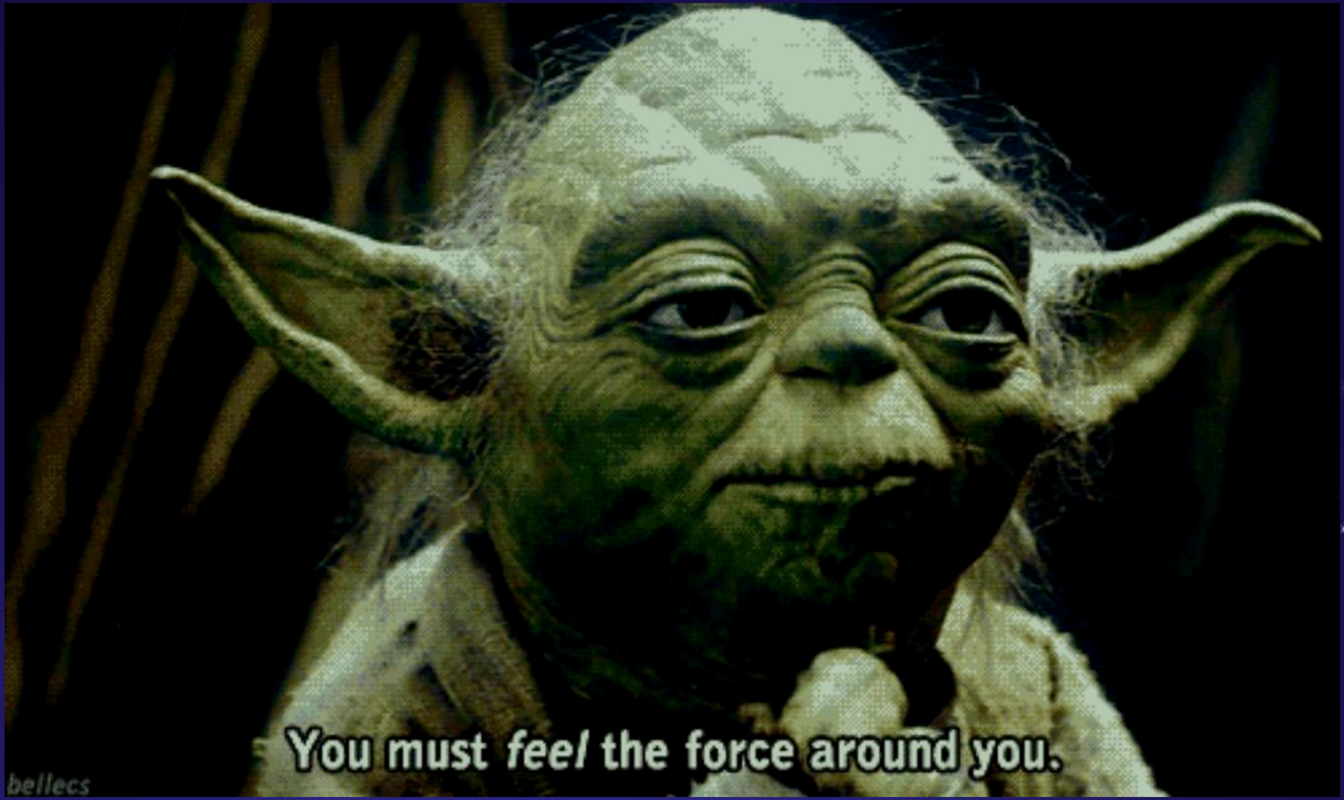
- Each supervised training example is a (language, structure) pair.
- There are only about 7,000 languages on Earth.
- Only about 60 languages on Earth are labeled (have treebanks).
- **Why Earth?**




JOHNS HOPKINS UNIVERSITY

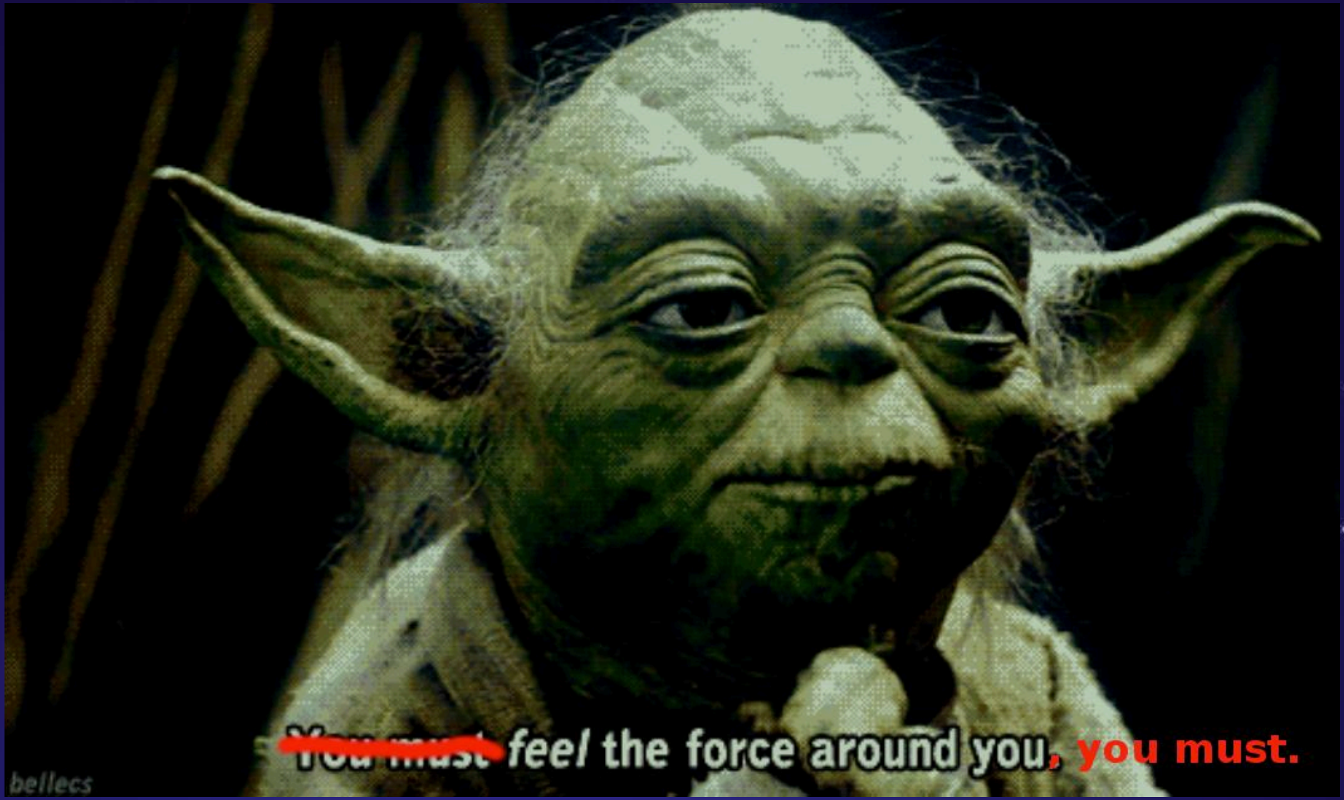
# Luckily

- We are not alone



# Luckily

- Not alone, we are



# We created …
# The Galactic Dependencies Treebanks!

- More than 50,000 synthetic languages!
  - Resemble real languages, but not found on Earth
- Each has a corpus of dependency parses
  - In the Universal Dependencies format
  - Vertices are words labeled with POS tags
  - Edges are labeled syntactic relationships
- Provide train/dev/test splits, alignments, tools

# Synthetic data elsewhere

- Computer Vision
  - Generating more data by rotating, enlarging….

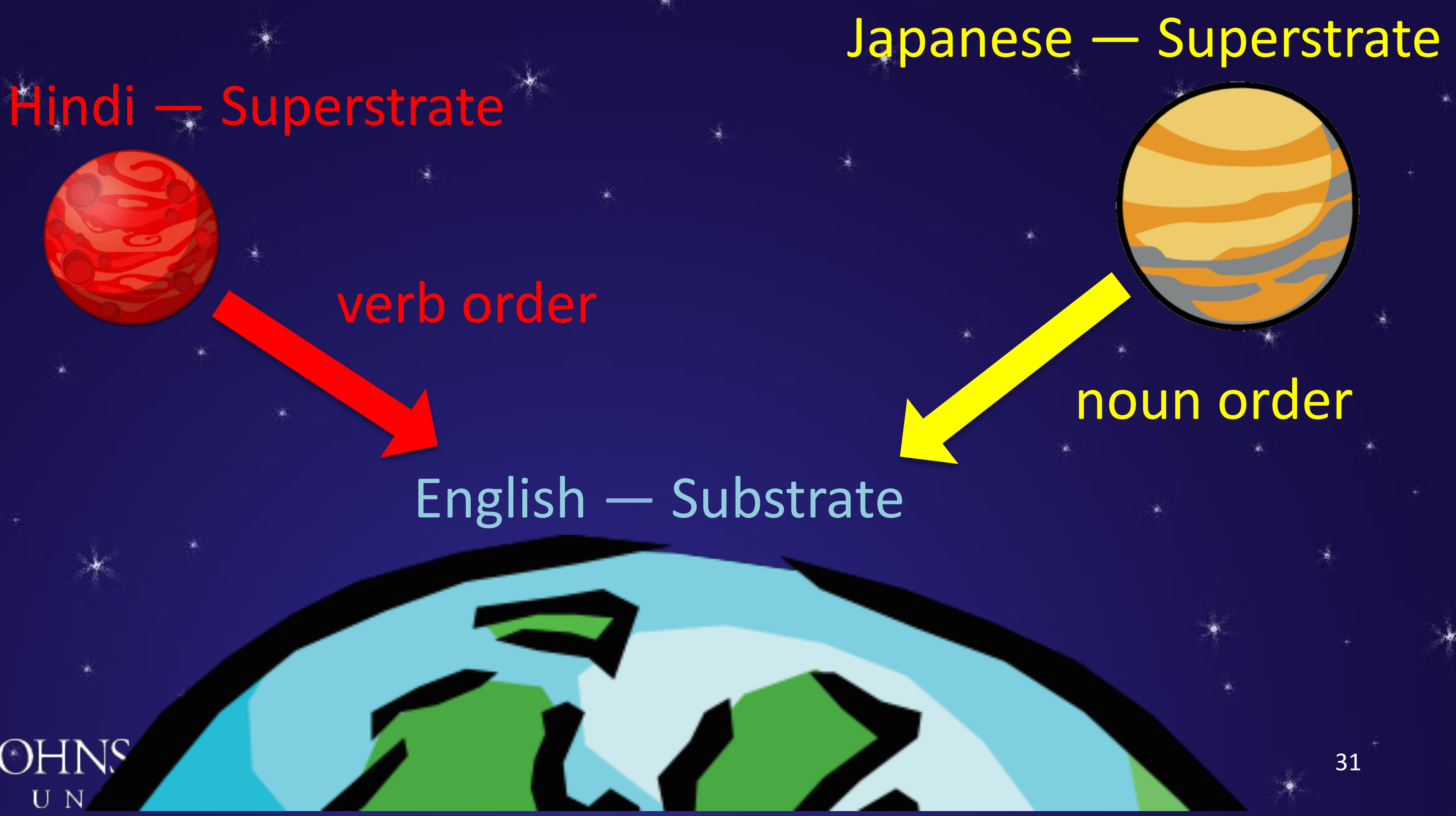

real synthetic variants

# Synthetic data elsewhere

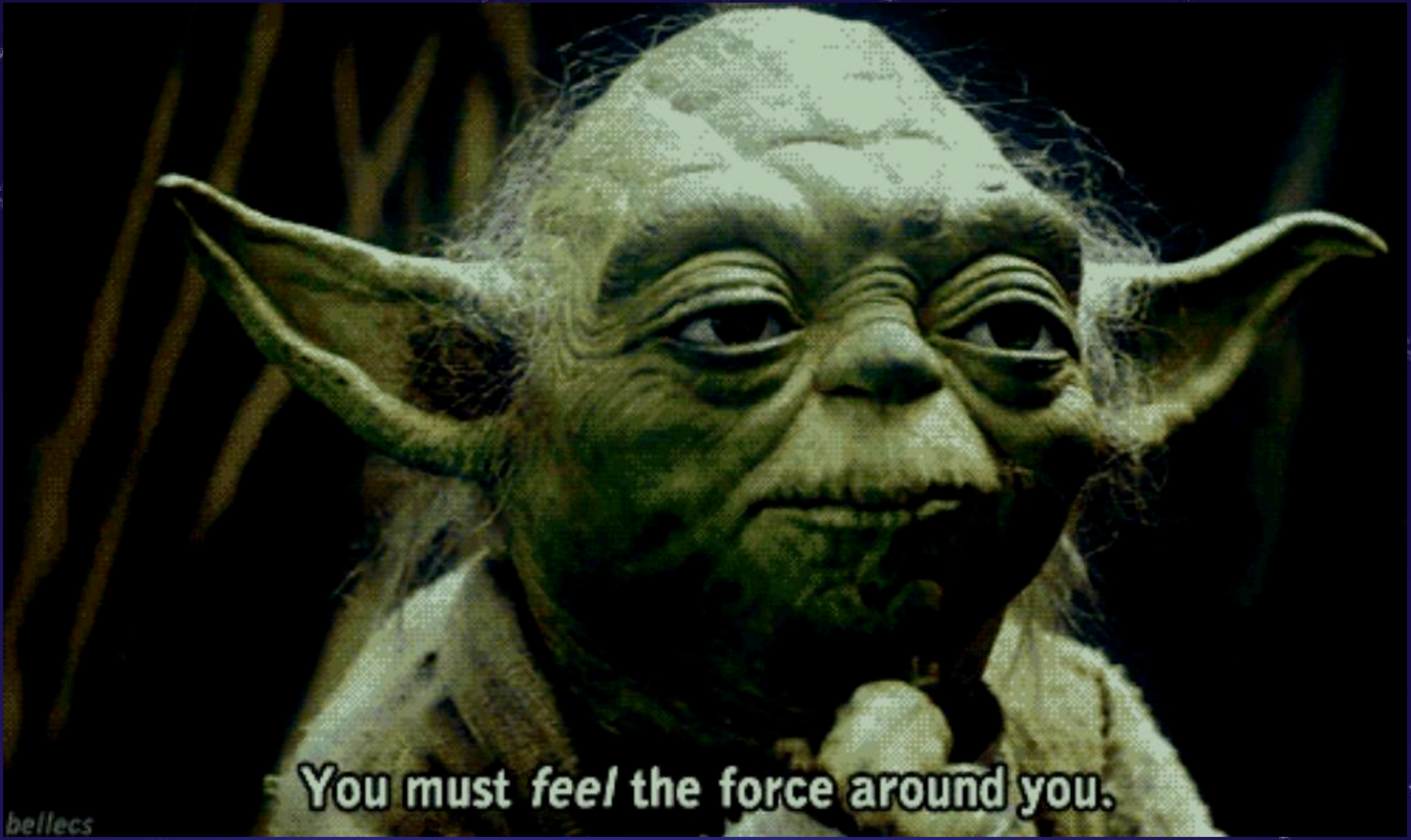
- Computer Vision
  - Generating more data by rotating, enlarging….
- Speech
  - Vocal Tract Length Perturbation (Jaitly and Hinton, 2013)
- NLP
  - bAbI (Weston et al., 2016)
  - The 30M Factoid Question-Answer Corpus (Serban et al., 2016)

# Substrate & Superstrates

(terms come from linguistics of creole languages)



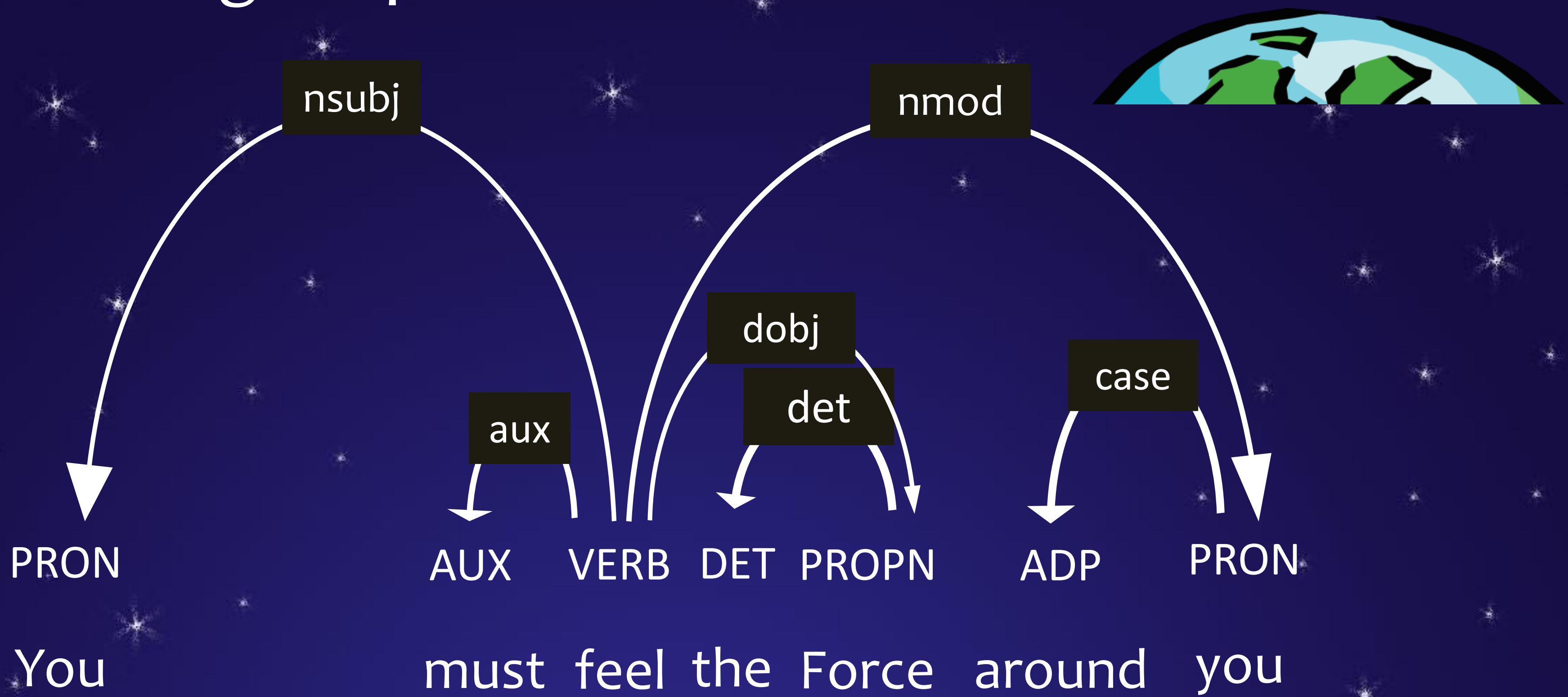
# You must feel the Force around you



# Example:

## You must feel the Force around you

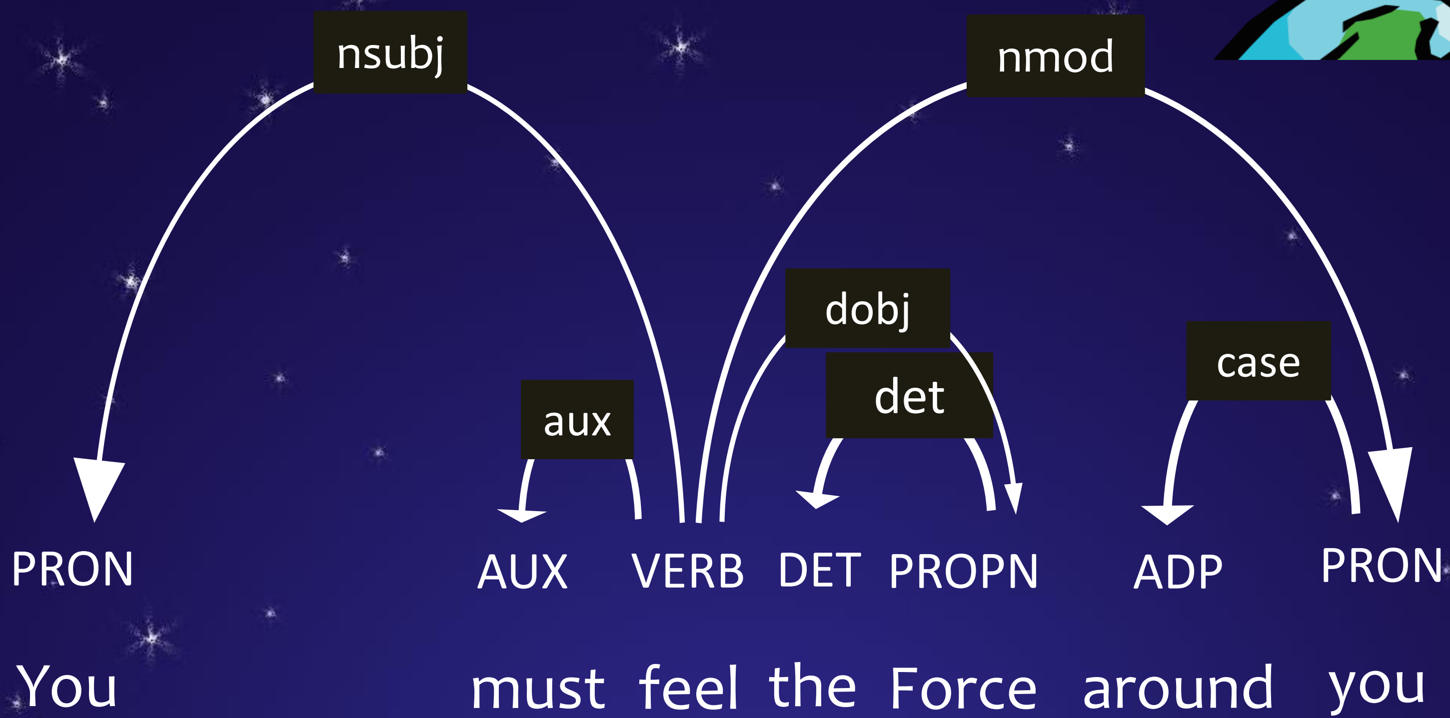
- A English parse:




Language: English

# Permute the children of verbs

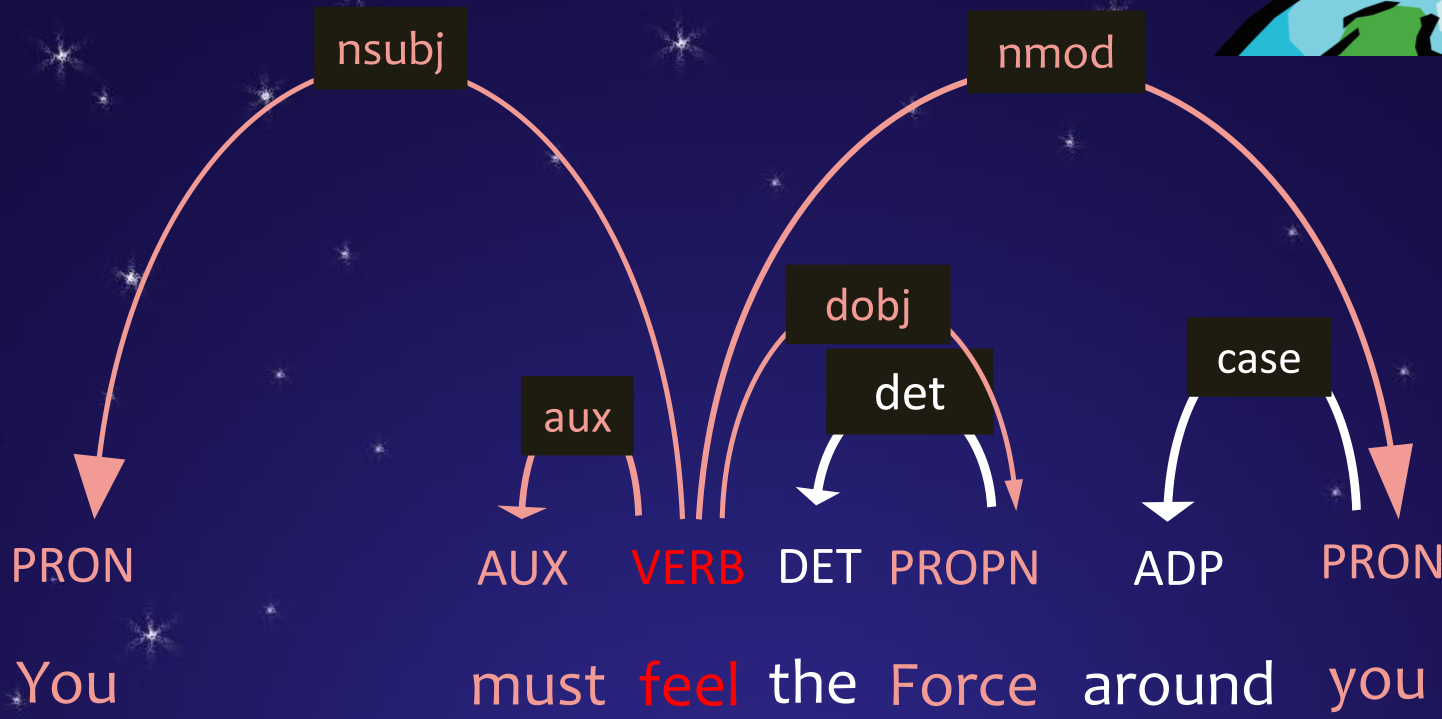
- SVO (English) → SOV (Hindi)




Language: English

# Permute the children of verbs

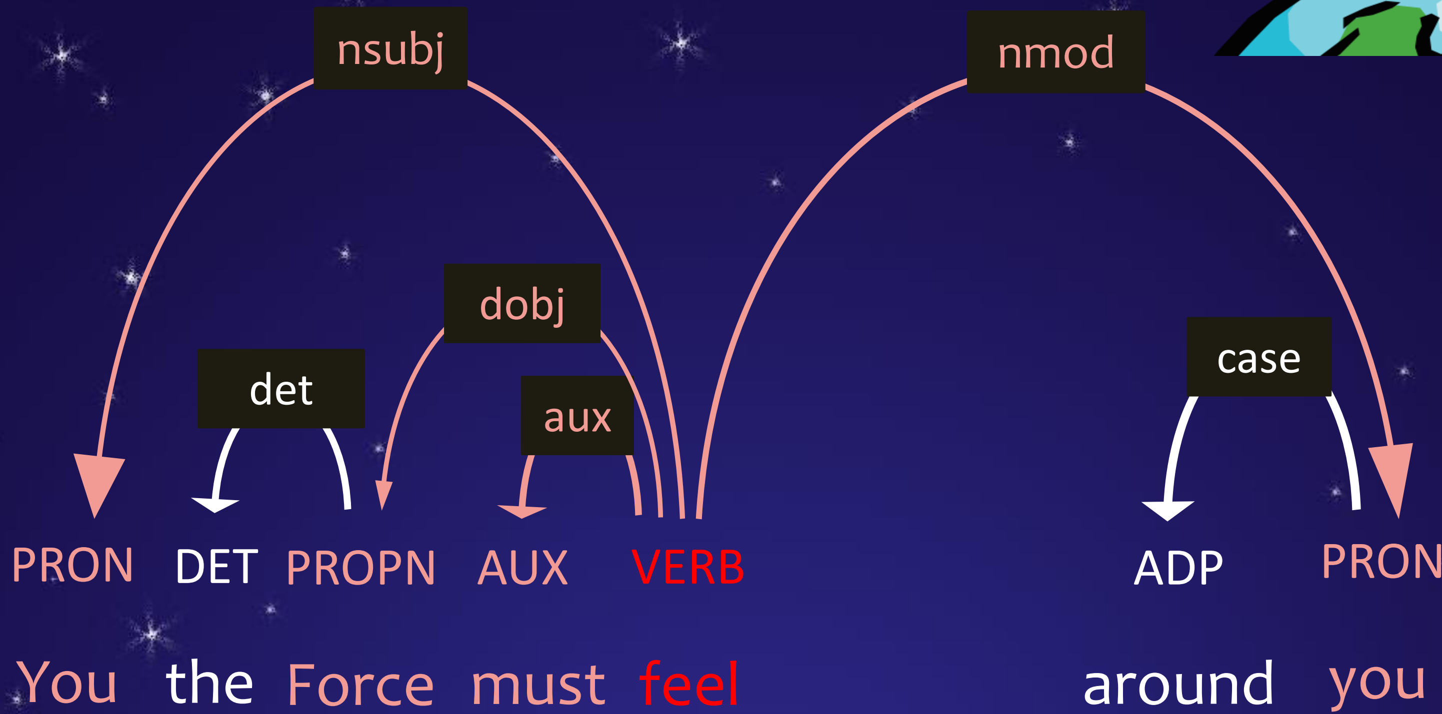
- SVO (English) → SOV (Hindi)




Language: English

# Permute the children of verbs

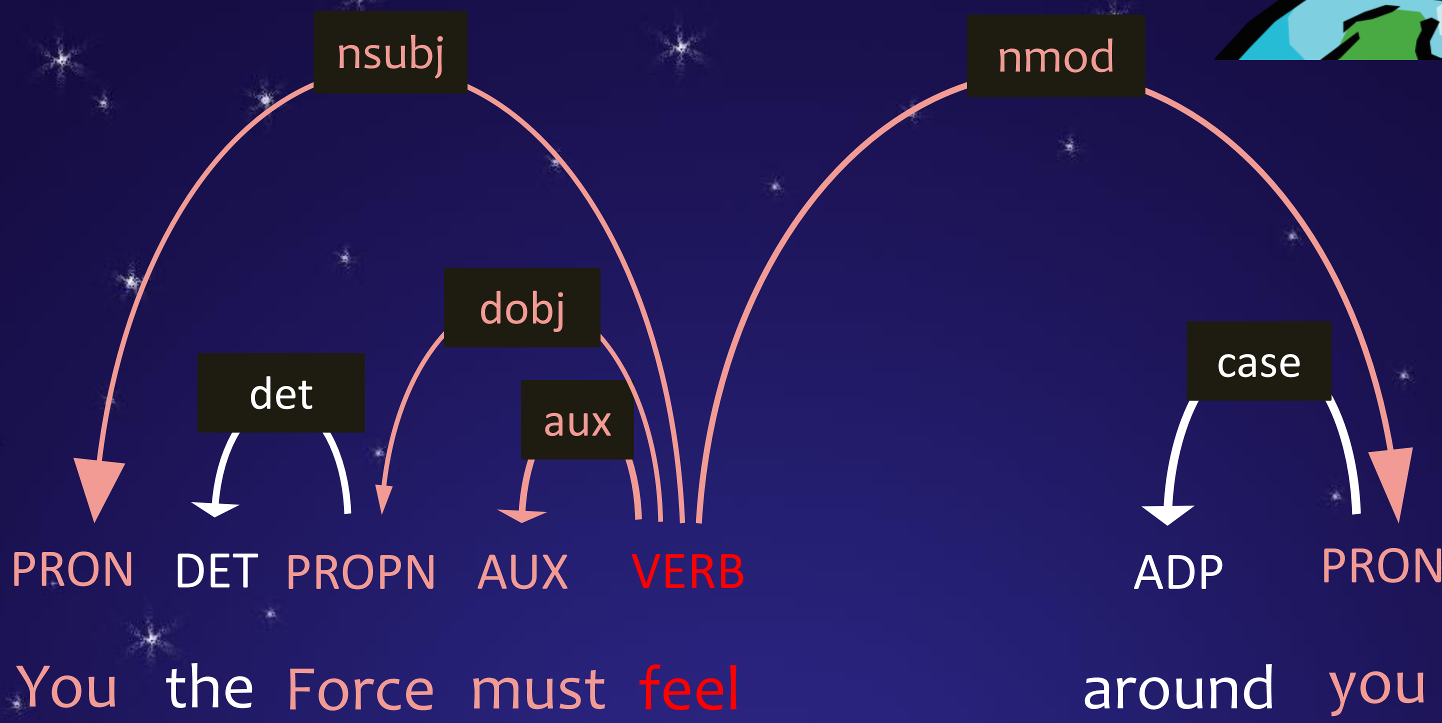
- SVO (English) → SOV (Hindi)




New language: English[Hindi/V]

# Permute the children of nouns

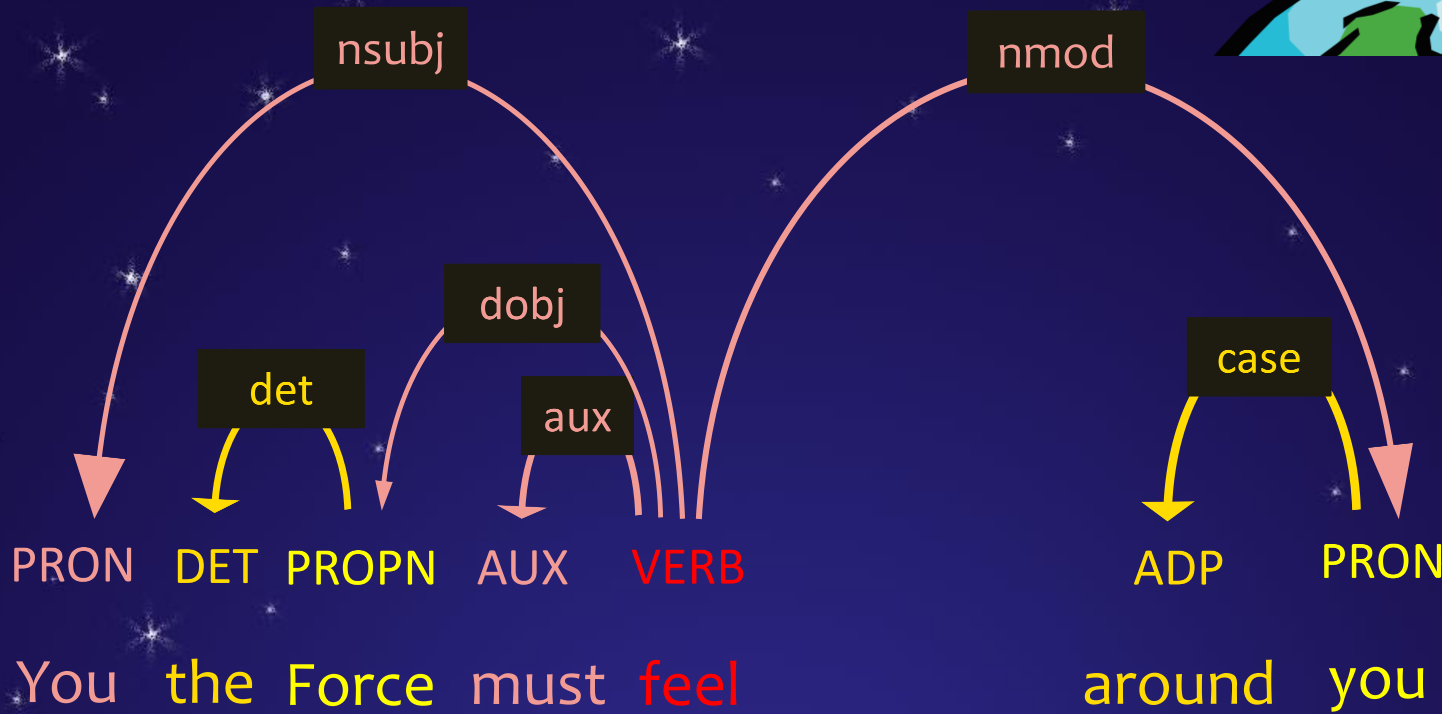
- Prepositions (English) → Postpositions (Japanese)




New language: English[Hindi/V]

# Permute the children of nouns

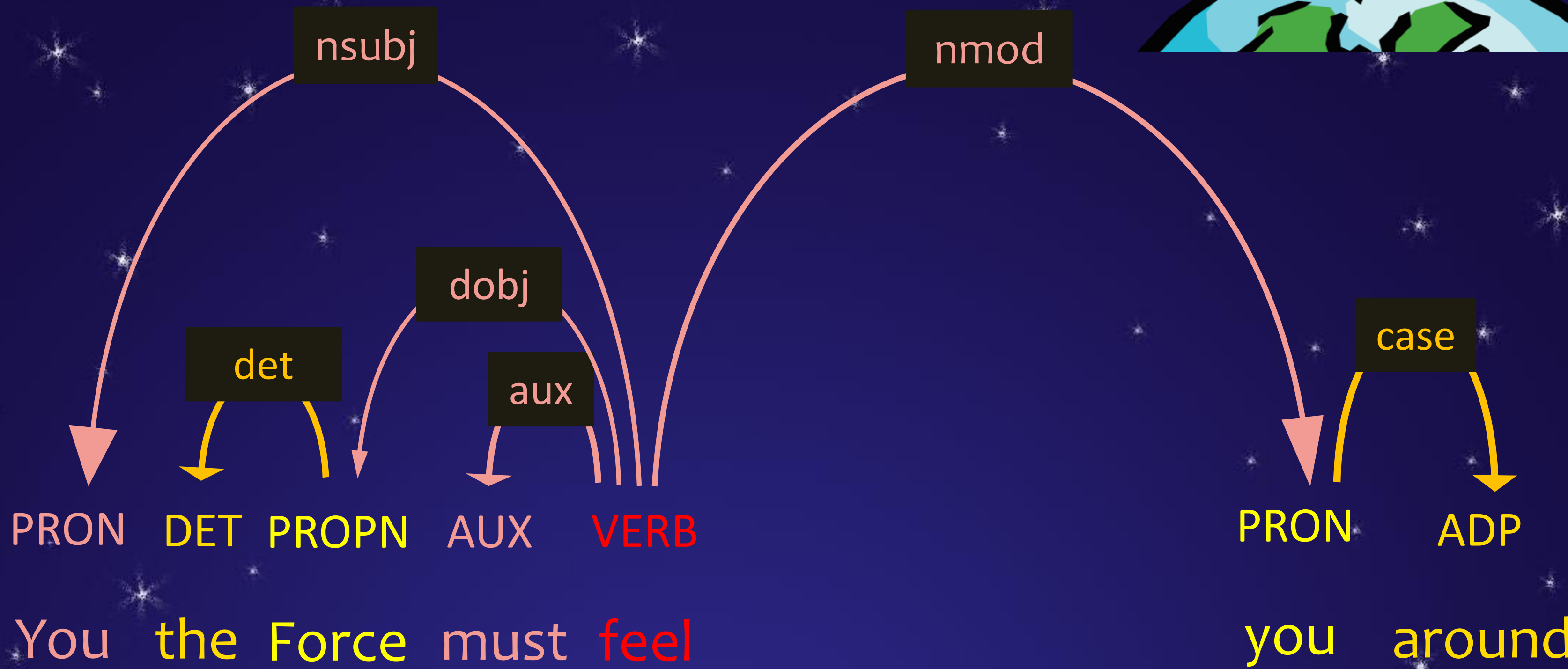
- Prepositions (English) → Postpositions (Japanese)




New language: English[Hindi/V]

# Permute the children of nouns

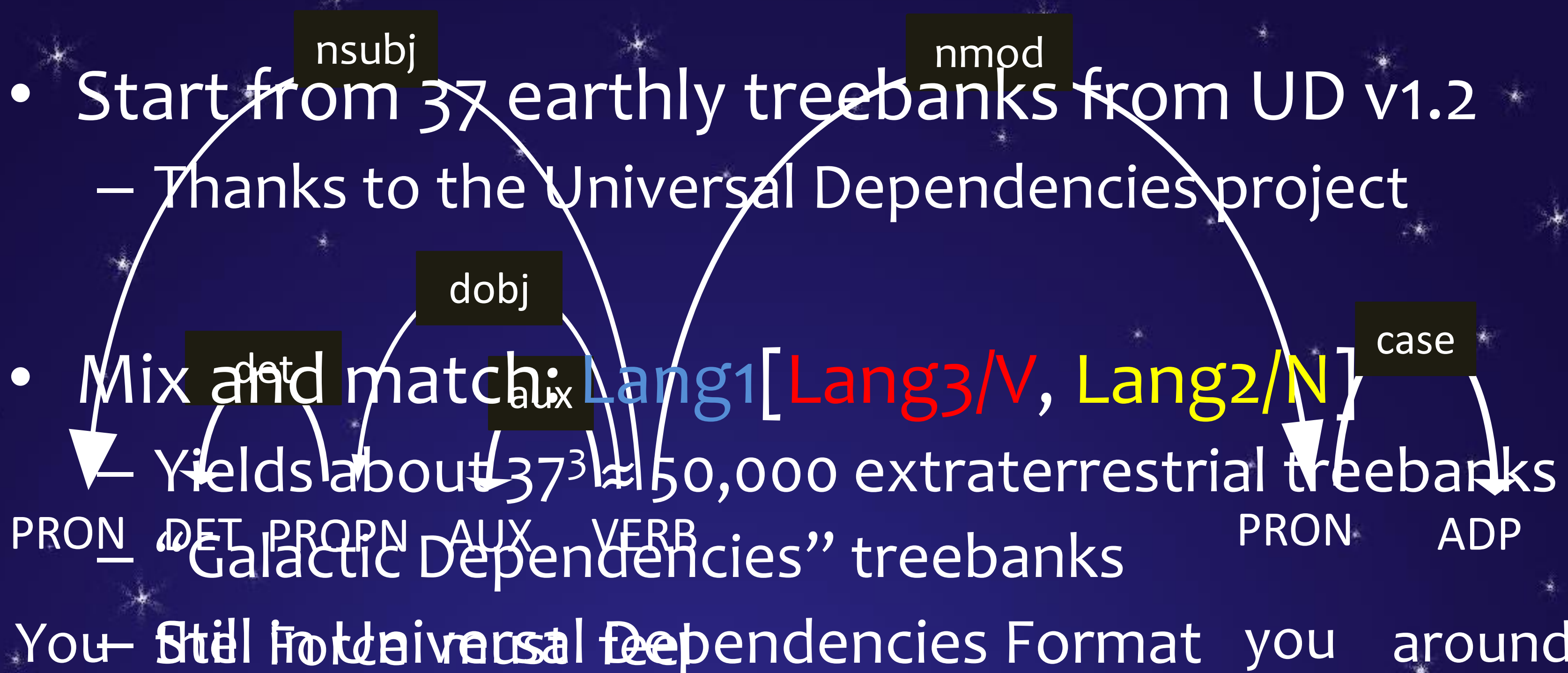
- Prepositions (English) → Postpositions (Japanese)



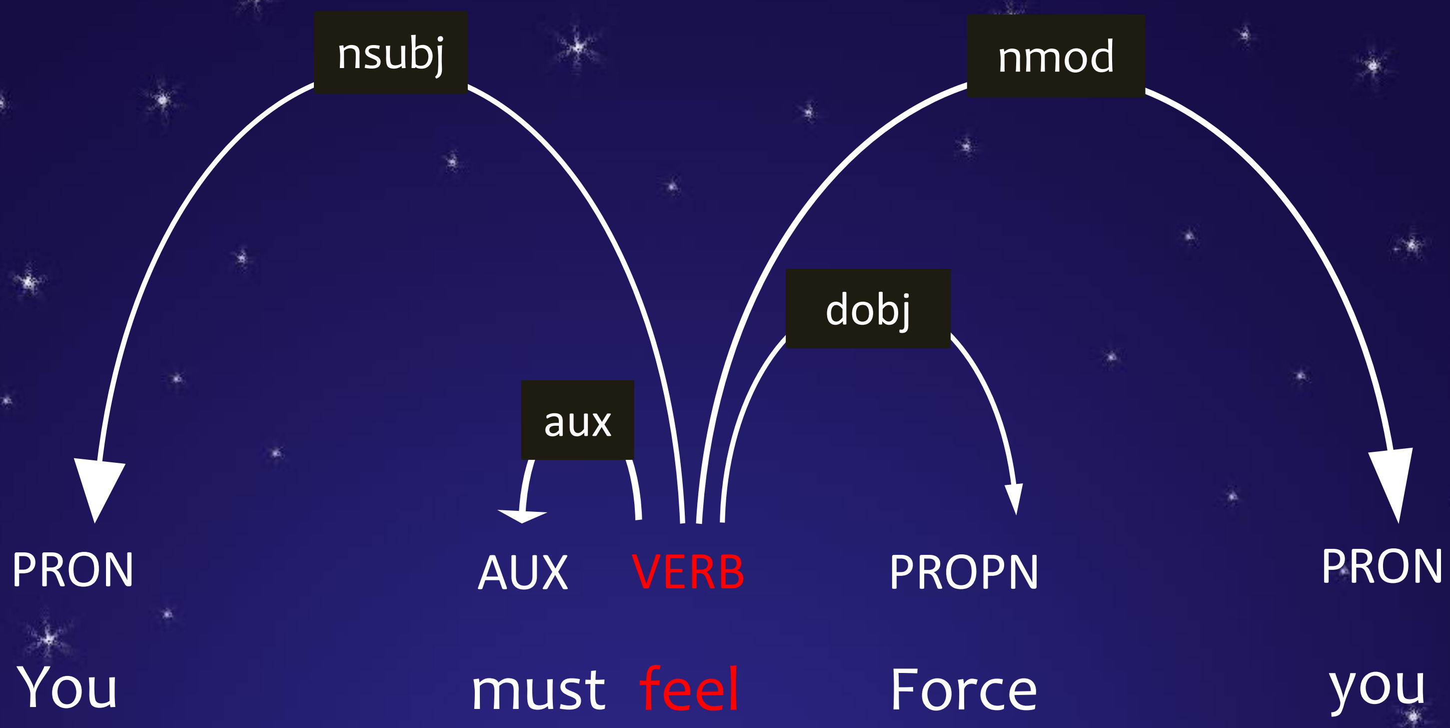

New language: English[Hindi/V, Japanese/N]

# What do we get?

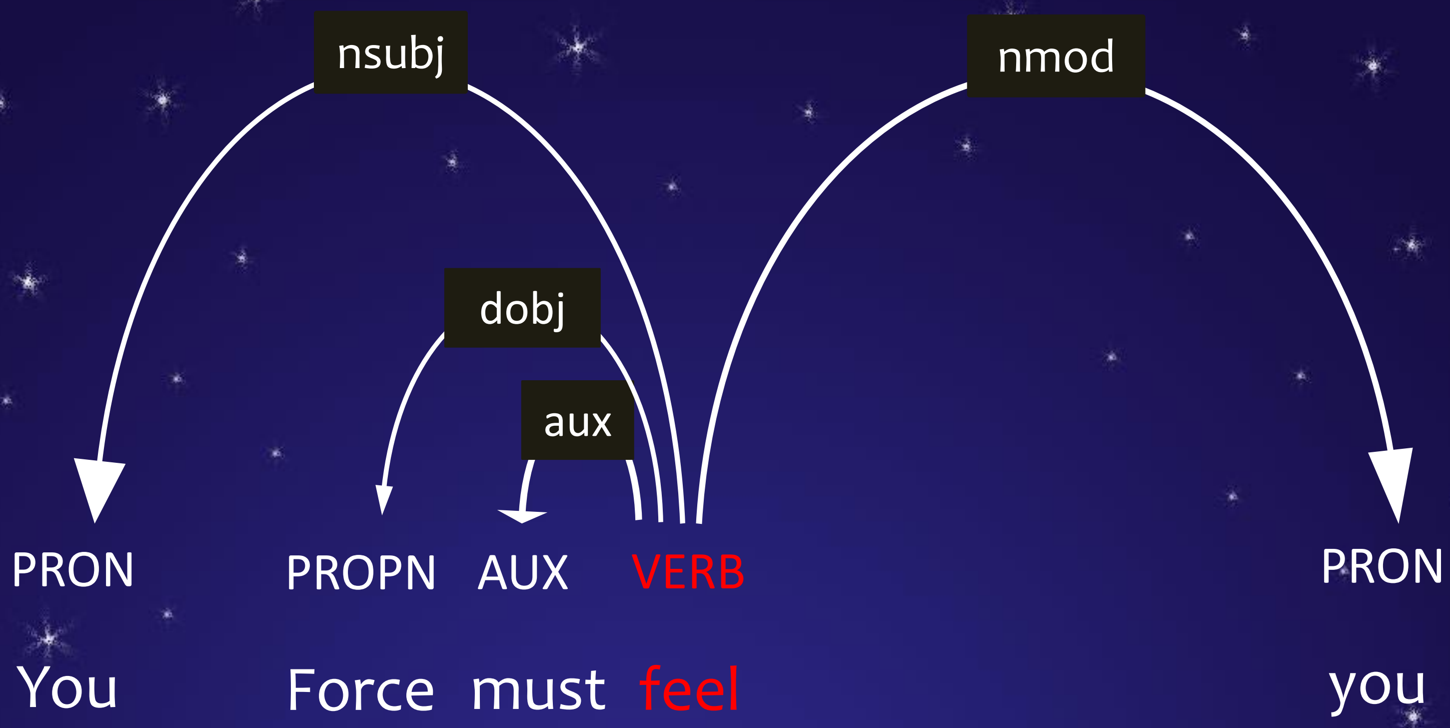
- New treebank: English[Hindi/V, Japanese/N]
- Start from 37 earthly treebanks from UD v1.2
  - Thanks to the Universal Dependencies project
- Mix and match: Lang1[Lang3/V, Lang2/N]
  - Yields about $37^3 \approx 50{,}000$ extraterrestrial treebanks
  - "Galactic Dependencies" treebanks
  - Still in Universal Dependencies Format



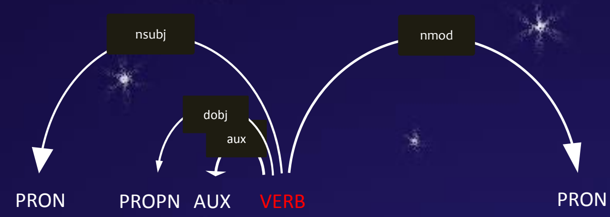
# How exactly do we permute?

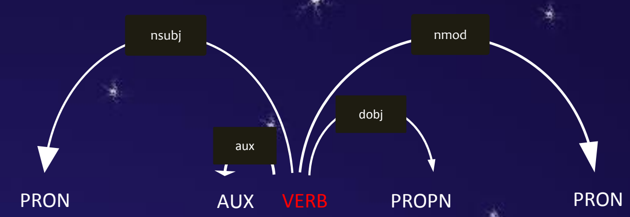


# SVO (English) → SOV (Hindi)



# Sampling

P(


|


, Hindi )

How many possible orders?

5!

# p( · | S V O , Hindi)

| Order | Prob. | BOS S adj. | S<V | O<V | SO adj. |
|---|---|---|---|---|---|
| S O V | | 1 | 1 | 1 | 1 |
| S V O | | 1 | 1 | 0 | 0 |
| O S V | | 0 | 1 | 1 | 1 |
| O V S | | 0 | 0 | 1 | 0 |
| V S O | | 0 | 1 | 0 | 1 |
| V O S | | 0 | 0 | 0 | 1 |

S:nsubj
O:dobj
V:VERB
BOS: Beginning of sentence

JOHNS HOPKINS UNIVERSITY

# p( · | S V O , Hindi)

| Order | Prob. | BOS S adj. | S<V | O<V | SO adj. |
|---|---|---|---|---|---|
| SOV | | 1 | 1 | 1 | 1 |
| SVO | | 1 | 1 | 0 | 0 |
| OSV | | 0 | 1 | 1 | 1 |
| OVS | | 0 | 0 | 1 | 0 |
| VSO | | 0 | 1 | 0 | 1 |
| VOS | | 0 | 0 | 0 | 1 |

- Each order has features as shown
- Train a log-linear model on Hindi treebank
- Sample from it to reorder English trees

S:nsubj
O:dobj
V:VERB
BOS: Beginning of sentence

# p( · | S V O , Hindi)

| Order | Prob. | BOS S adj. | S<V | O<V | SO adj. |
|---|---|---|---|---|---|
| S O V | 0.8 | 1 | 1 | 1 | 1 |
| S V O | 0.03 | 1 | 1 | 0 | 0 |
| O S V | 0.1 | 0 | 1 | 1 | 1 |
| O V S | 0.02 | 0 | 0 | 1 | 0 |
| V S O | 0.03 | 0 | 1 | 0 | 1 |
| V O S | 0.02 | 0 | 0 | 0 | 1 |

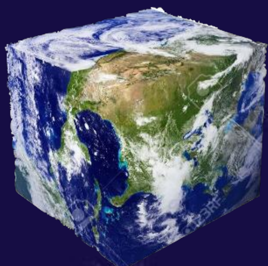
S:nsubj
O:dobj
V:VERB
BOS: Beginning of sentence

# Are Synthetic Languages Useful??

- The languages should be diverse enough.
- The languages should be in the galaxy (in-domain)!

# Evaluation: Parsability

- Is this language functional enough to survive during human/alien evolution?

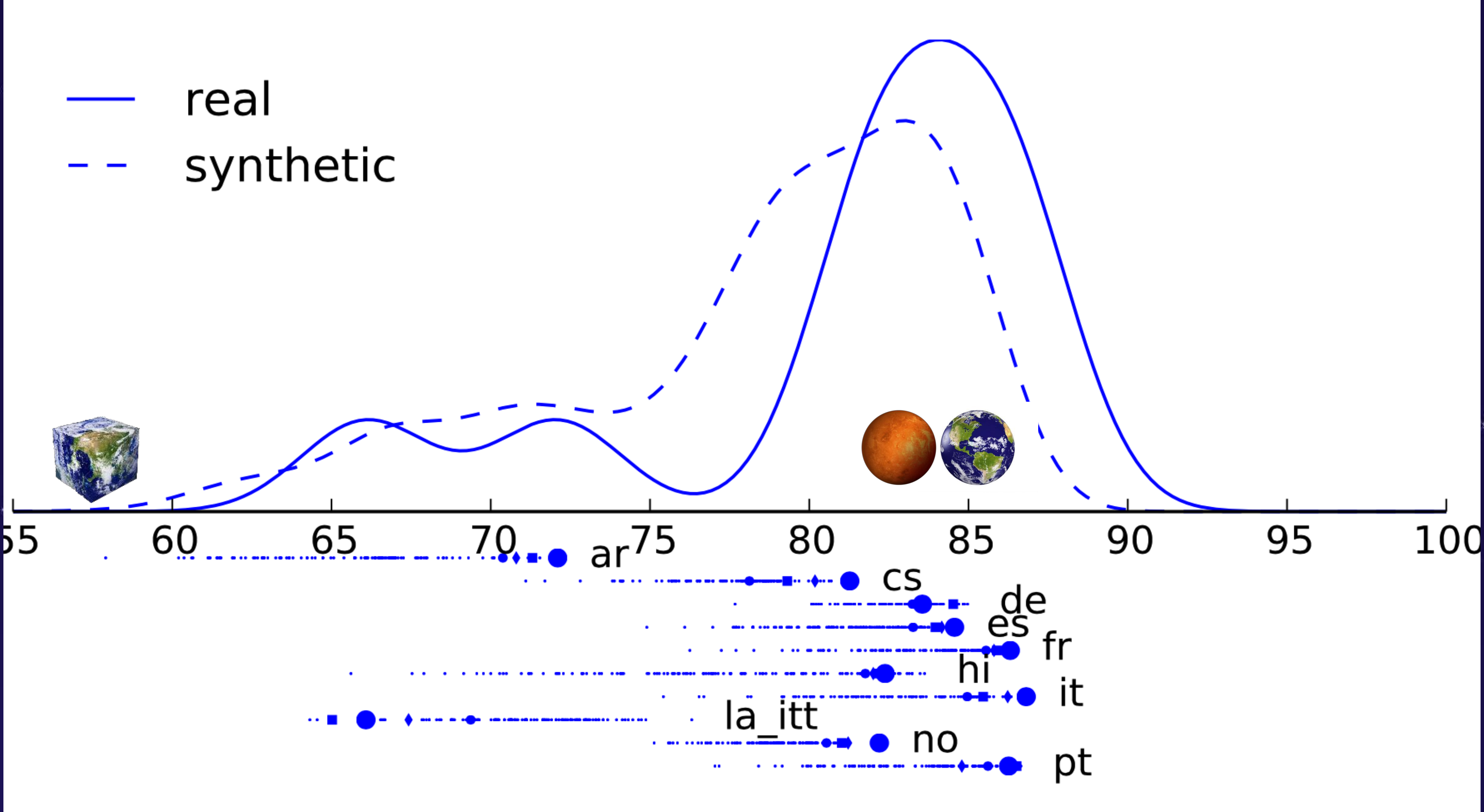


- Less parsable → worse for communication

- Train a parser on some trees of the language
- Evaluate UAS (unlabeled attachment score) on held-out trees of the same language

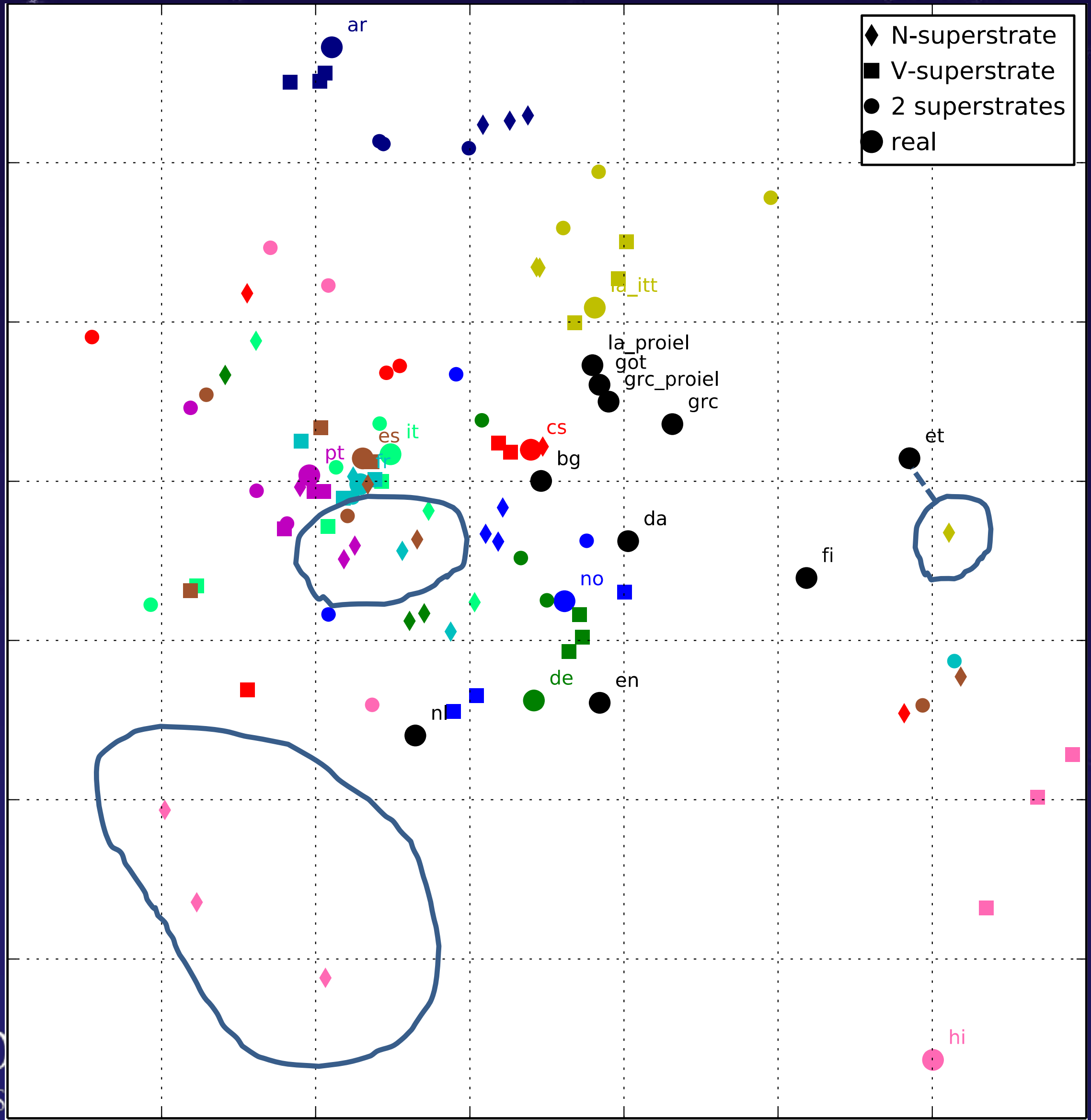
JOHNS HOPKINS UNIVERSITY

# Evaluation: Parsability



distance ≈ parsing transfer accuracy

# Evaluation: Diversity

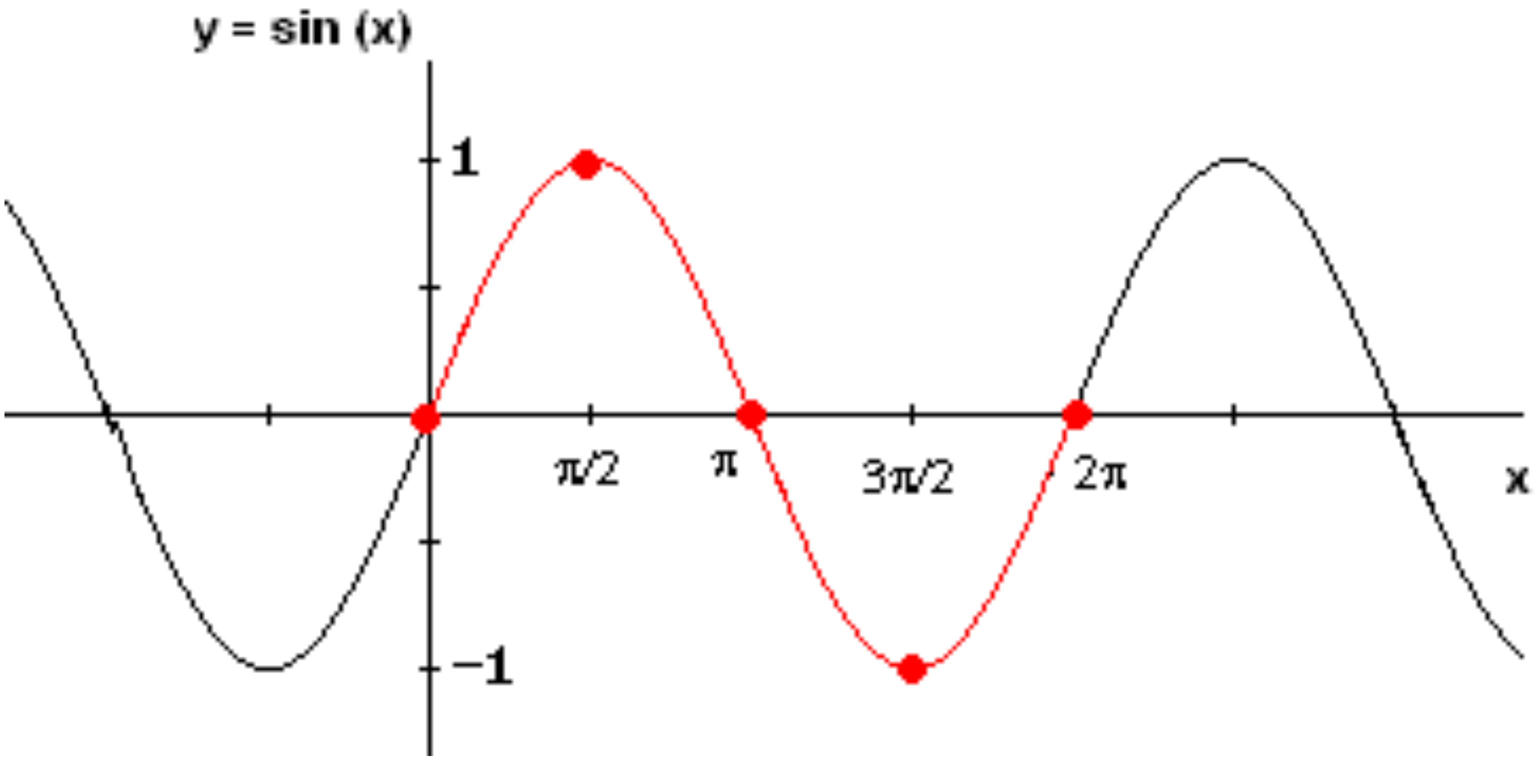



JOHNS HOPKINS UNIVERSITY

Stats works backward from data to parameters.

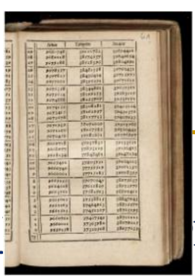
So it's like function inversion … so,

# How do you compute $\sin^{-1}(y)$?

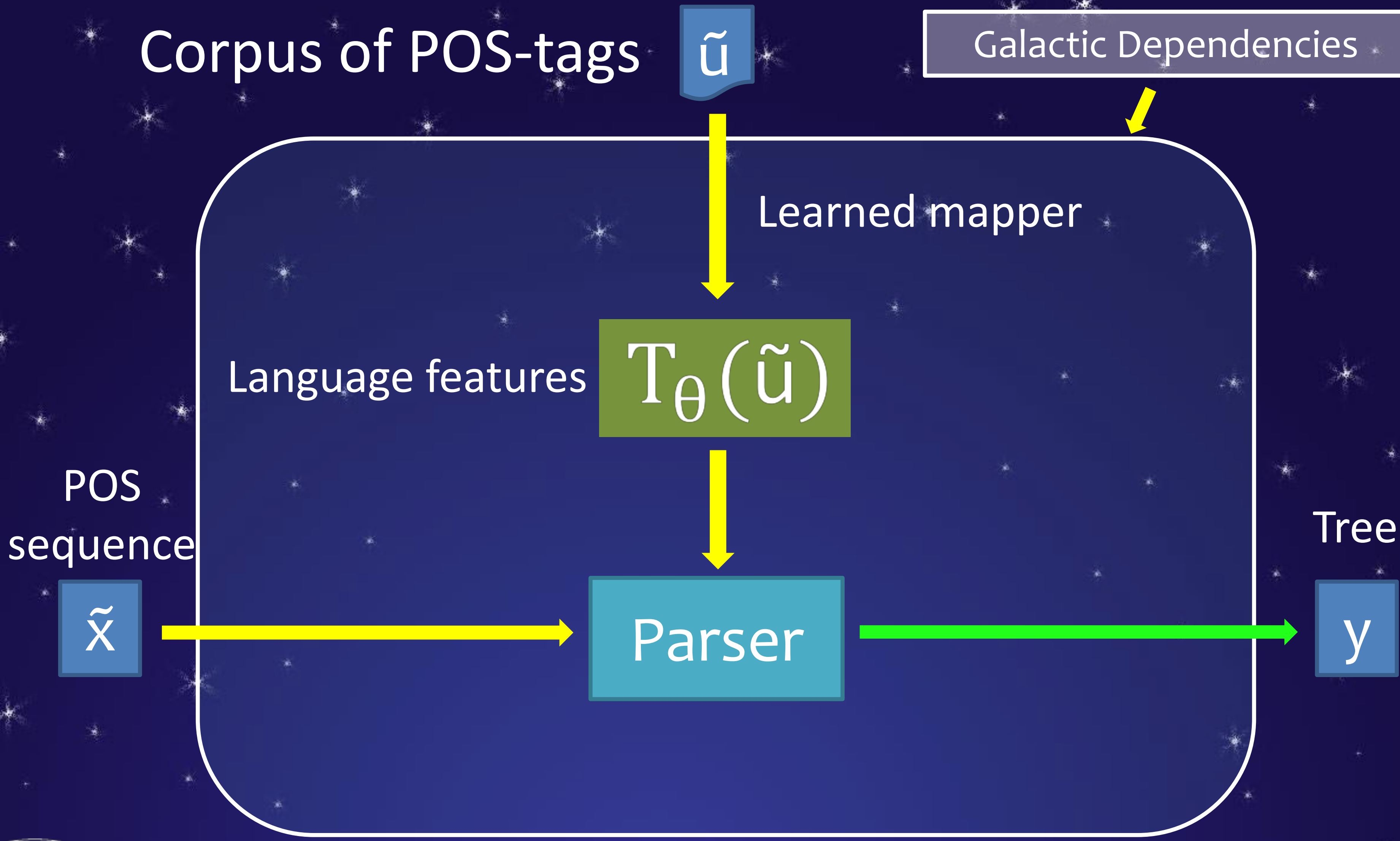


# How do you compute $\sin^{-1}(y)$?

- Method 1: Use the Taylor series formula.
  - In ML: Such closed-form methods occasionally exist for fitting a model, e.g., spectral learning of HMMs.
- Method 2: Local search for a root of y=sin(x).
  - In ML: This is what we usually do! Sadly, no multi-dim bisection method exists: use EM, MAP, HMC, etc.
- Method 3: **Precompute** a table of all (x,sin(x)), look up rows with sin(x) ≈ y, and interpolate.
  - In ML: This work! "Scattershot" precomputation of many reasonable observed sentences, and then learn how to work backwards from y to x …



neural net grammar corpus



Corpus of POS-tags
ũ
Galactic Dependencies
Learned mapper
Language features
$T_\theta(\tilde{u})$
POS
sequence
x̃
Parser
Tree
y

Corpus of POS-tags $\tilde{u}$

Galactic Dependencies

Learned mapper

Language features $T_\theta(\tilde{u})$

POS sequence $\tilde{x}$

Parser

Tree $y$

# Wang and Eisner (2017)

Corpus of POS-tags $\tilde{u}$

Galactic Dependencies

Learned mapper

Language features $T_\theta(\tilde{u})$

Fine-grained syntactic typology

# Prediction of Syntactic Typology

Corpus of POS-tags $\tilde{u}$

Galactic Dependencies

Learned mapper

Language features $T_\theta(\tilde{u})$

# Prediction of Syntactic Typology

Corpus of POS-tags $\tilde{u}$

Galactic Dependencies

Learned mapper

Language features $T_\theta(\tilde{u})$

Trained with supervision!

# Supervised Training

POS-corpus → ũ

Vector of length 57 → True Typology

Language 1 ( 

- PRON AUX …
- VERB PROPN …

…

,

| | | | | |
|---|---|---|---|---|
| 0.1 | 0.8 | 0.1 | 0.1 | … |

)

Language 2 (

- VERB NOUN…
- NOUN DET…
- NOUN ADJ …

…

,

| | | | | |
|---|---|---|---|---|
| 0.1 | 0.2 | 0.8 | 0.1 | … |

)

.
.
.

$$\text{Directionality}(r) = \frac{\#\, r\ from\ left\ to\ right}{\#\, r}$$

# Prediction of Syntactic Typology

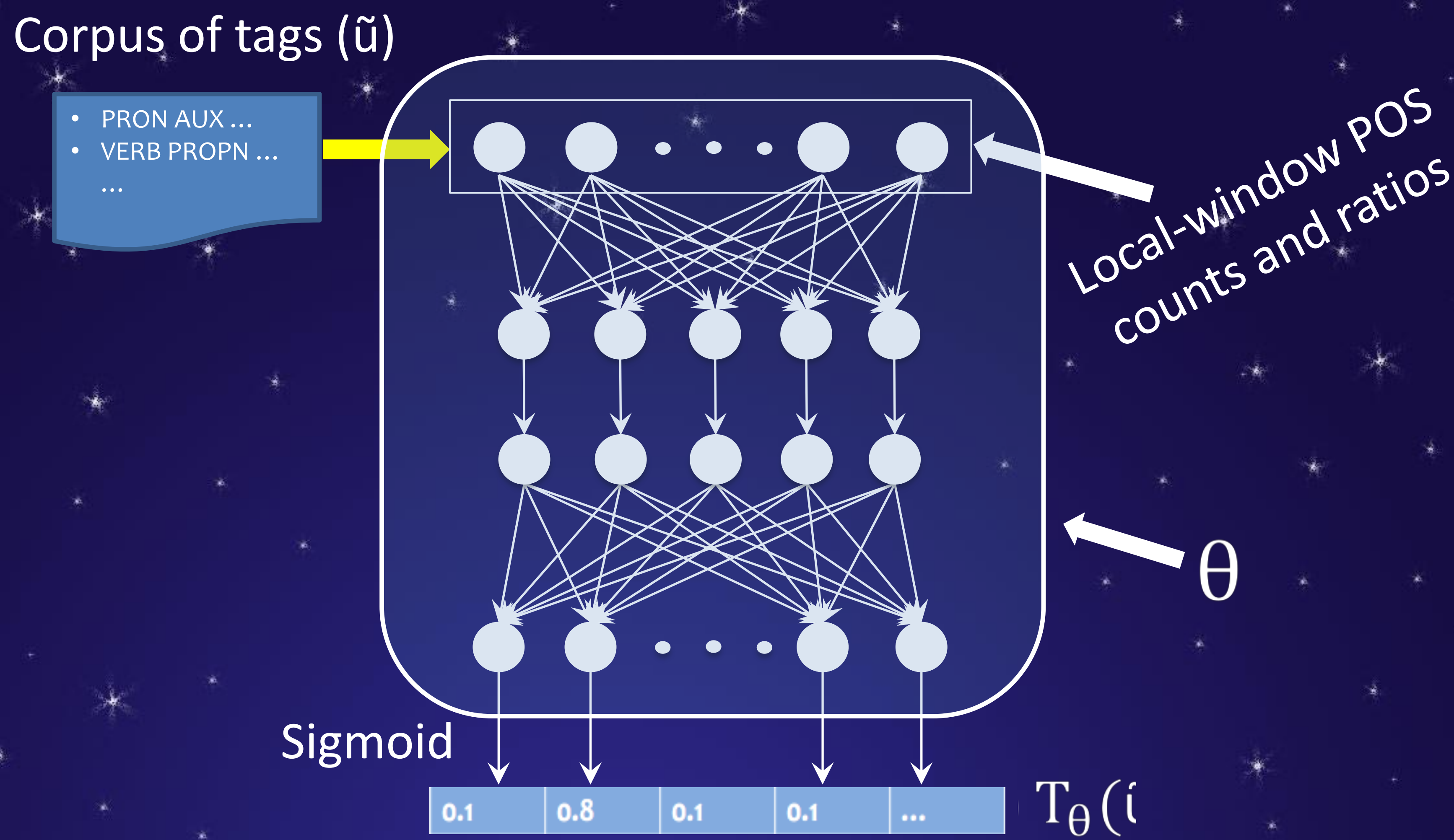
Corpus of POS-tags ũ

Galactic Dependencies

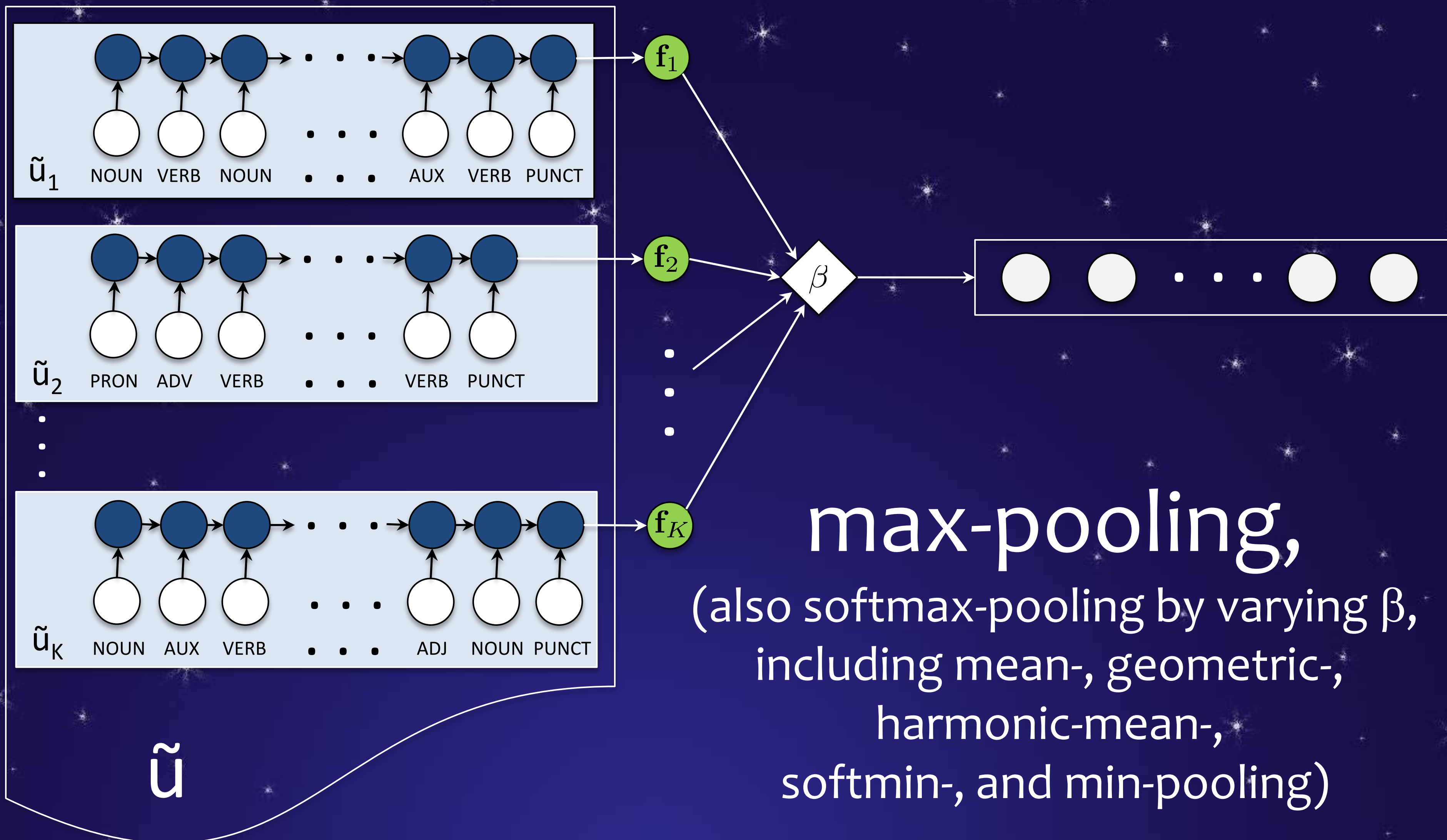
Learned mapper

Language features $\theta(\tilde{u})$

# Hand-designed features



# LSTM features




max-pooling,

(also softmax-pooling by varying β, including mean-, geometric-, harmonic-mean-, softmin-, and min-pooling)

# Data

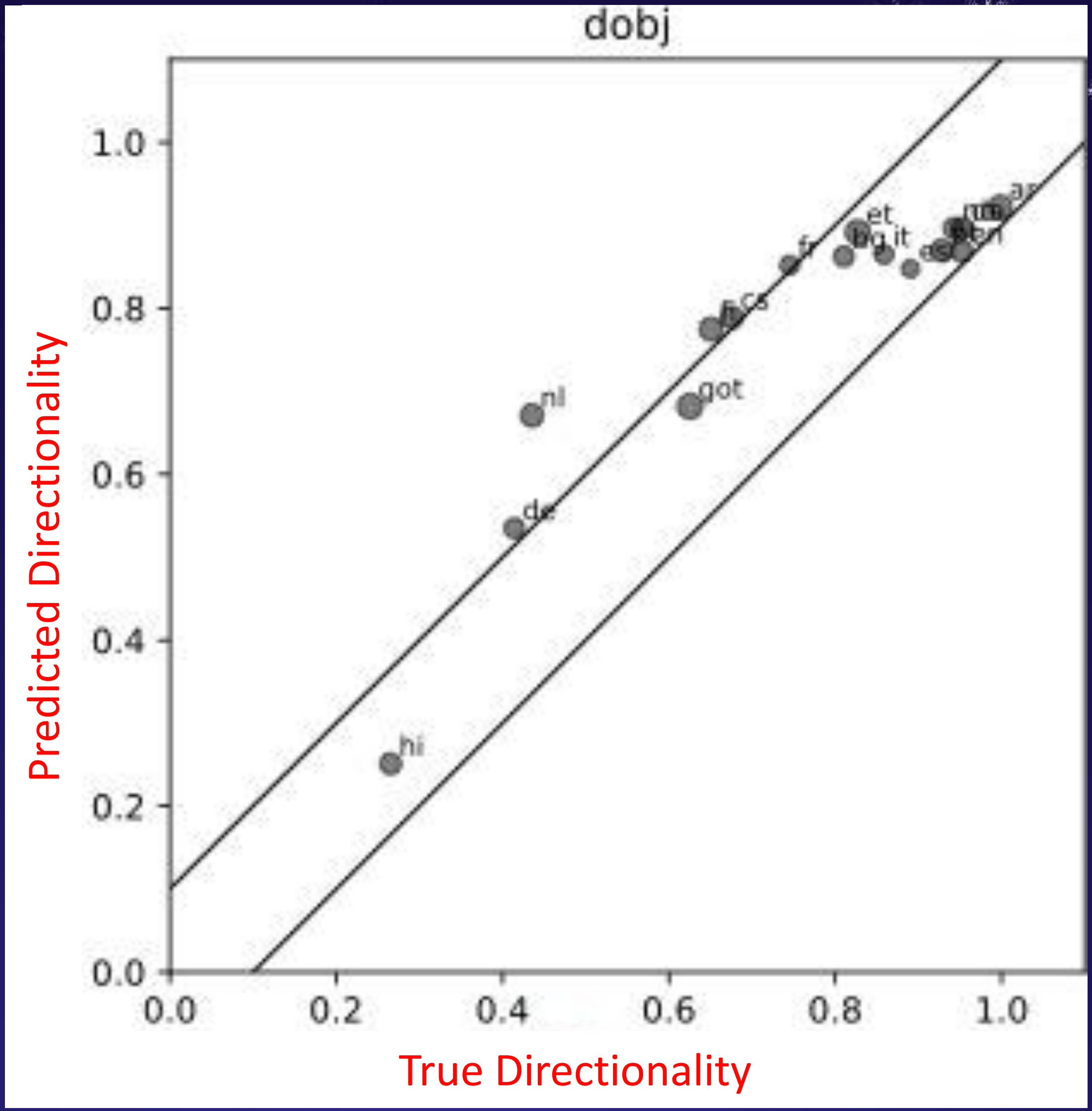
- Universal Dependencies version 1.2
  - A collection of 37 dependency treebanks for 33 languages
- Galactic Dependencies version 1.0

  UD: 20 treebanks drawn from above

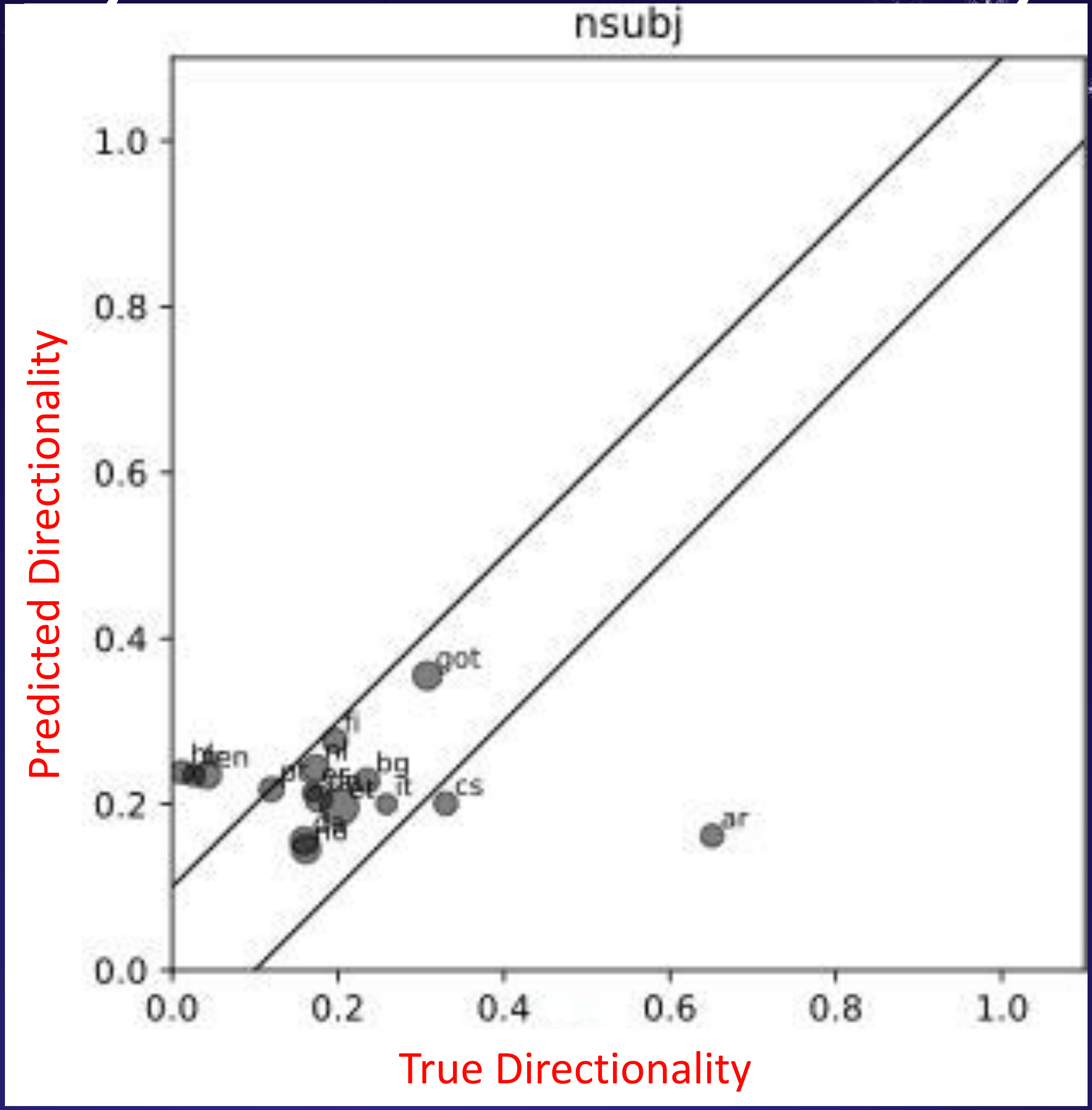
  \+ GD: about 8000 treebanks by mix-and-match

| Train | Test |
|---|---|
| cs, es, fr, hi, de, it, la itt, no, ar, pt en, nl, da, fi, got, grc, et, la proiel, grc proiel, bg | hr, ga, he, hu, fa, ta, cu, el, ro, sl, ja ktc, sv, id, eu, pl |

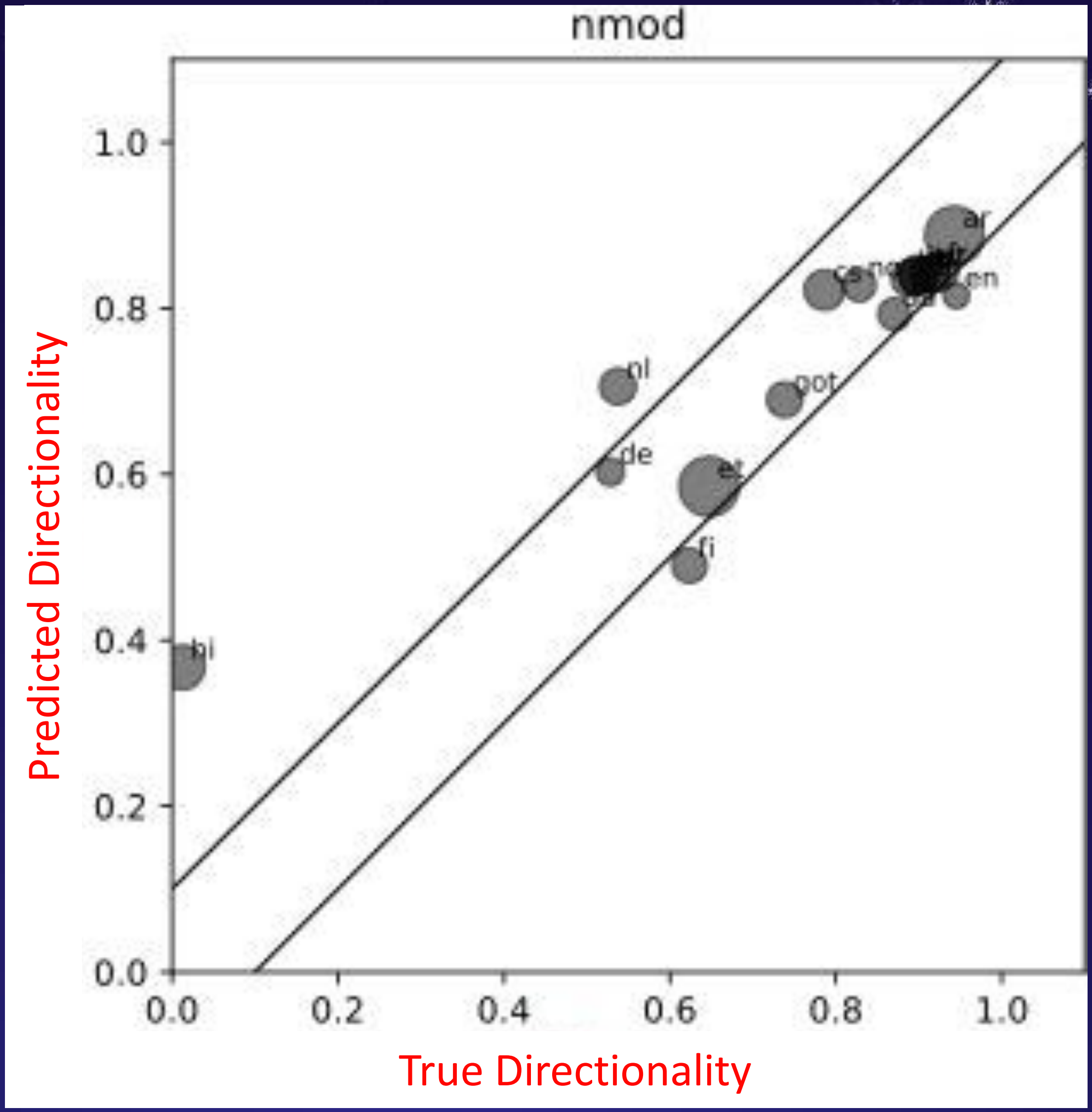
# dobj: Head Verb -> Direct Object



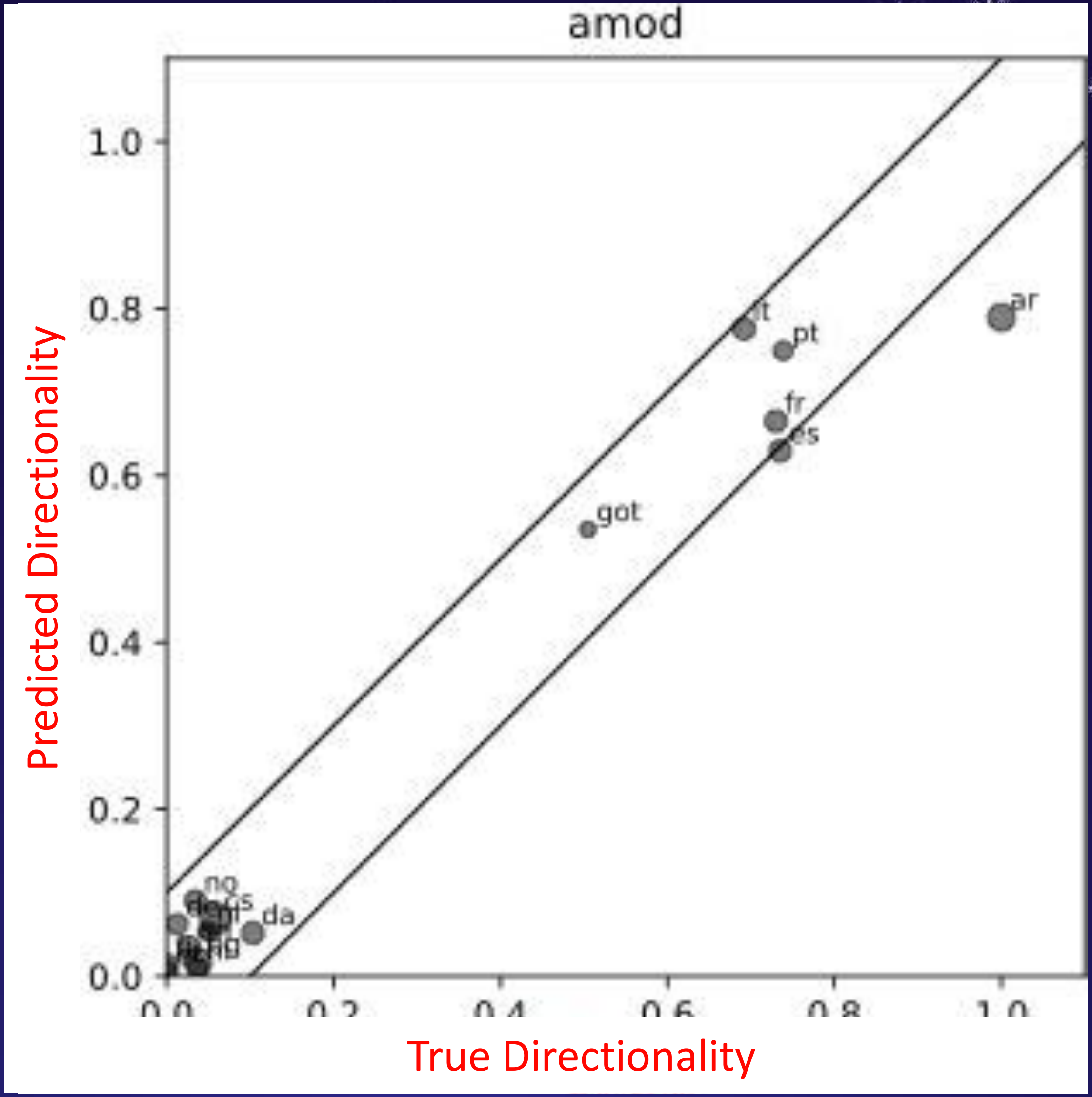
# nsubj: Head Verb -> Subject



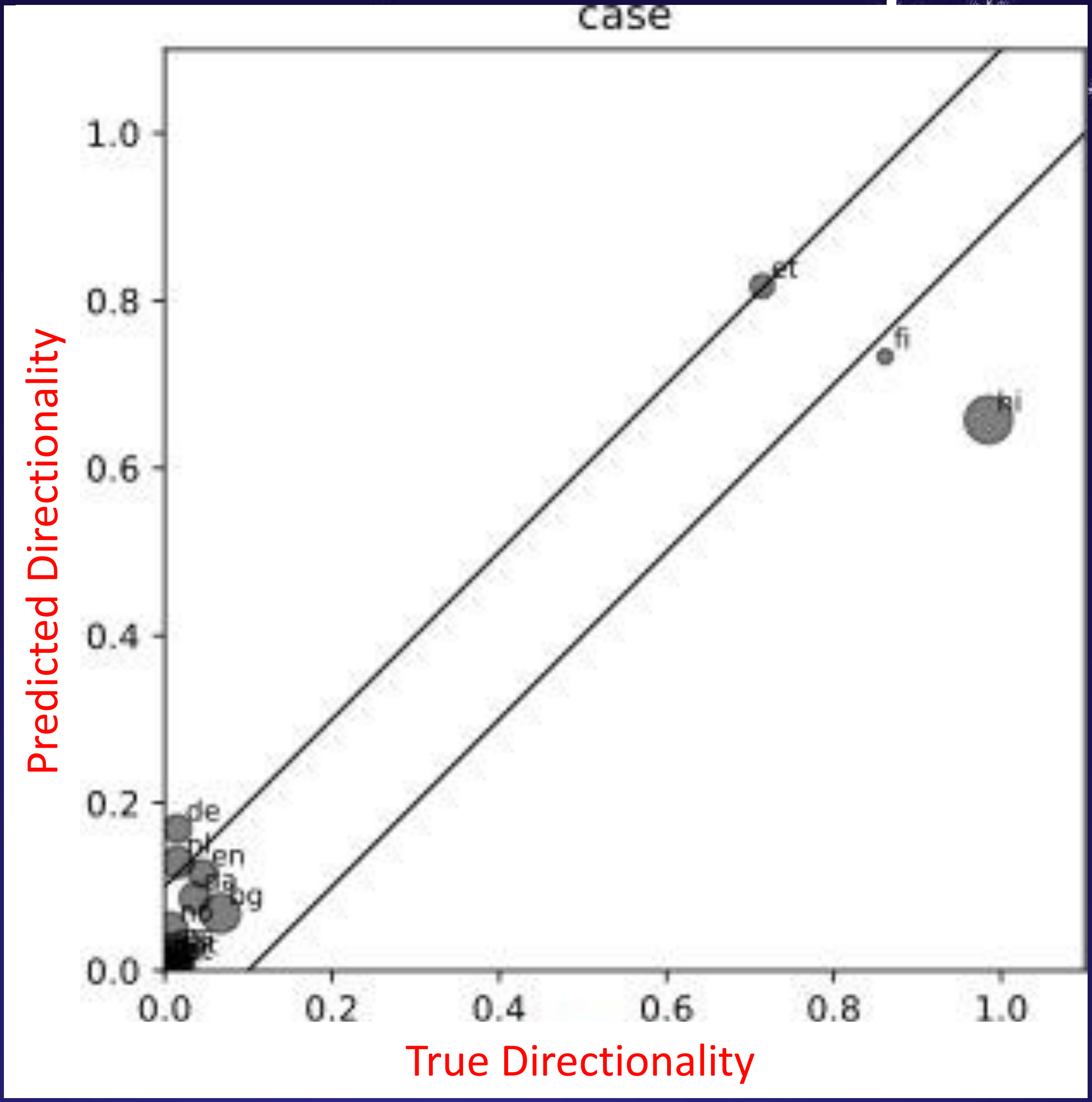
# nmod: Head Noun -> Nominal Modifier



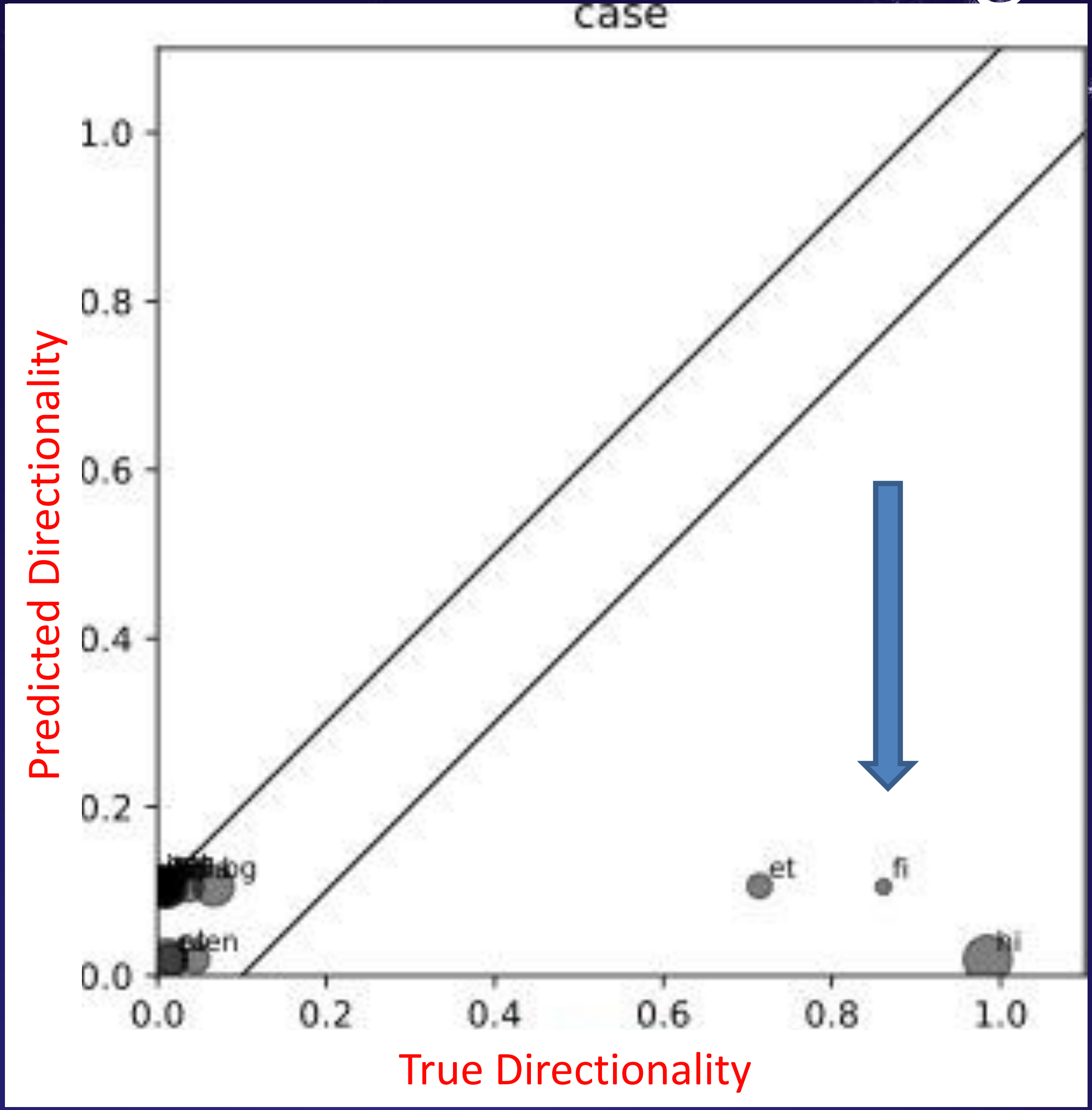
# amod: Head Noun -> Adjectival Modifier



# case: Head Noun -> Adposition

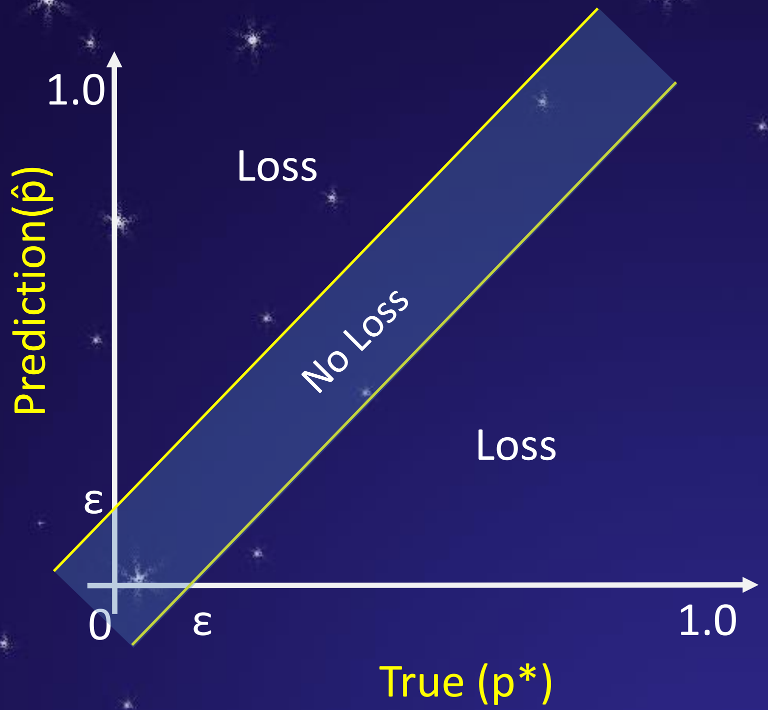


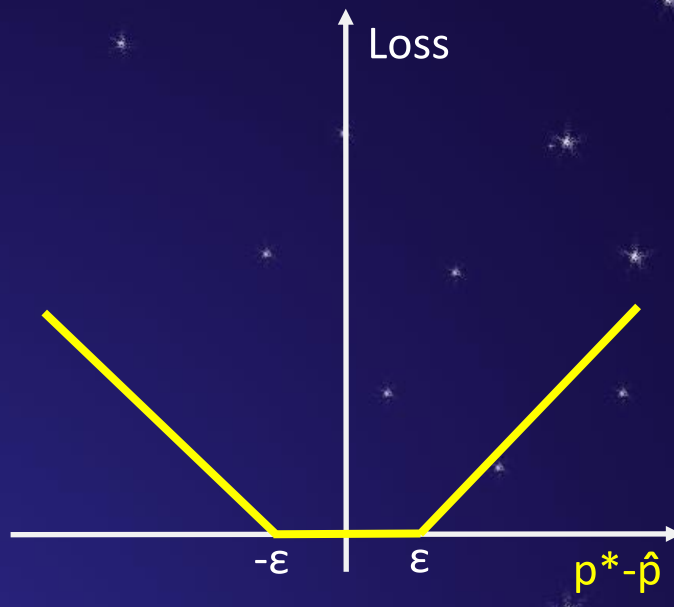
# case (Trained on 16 Real Languages)



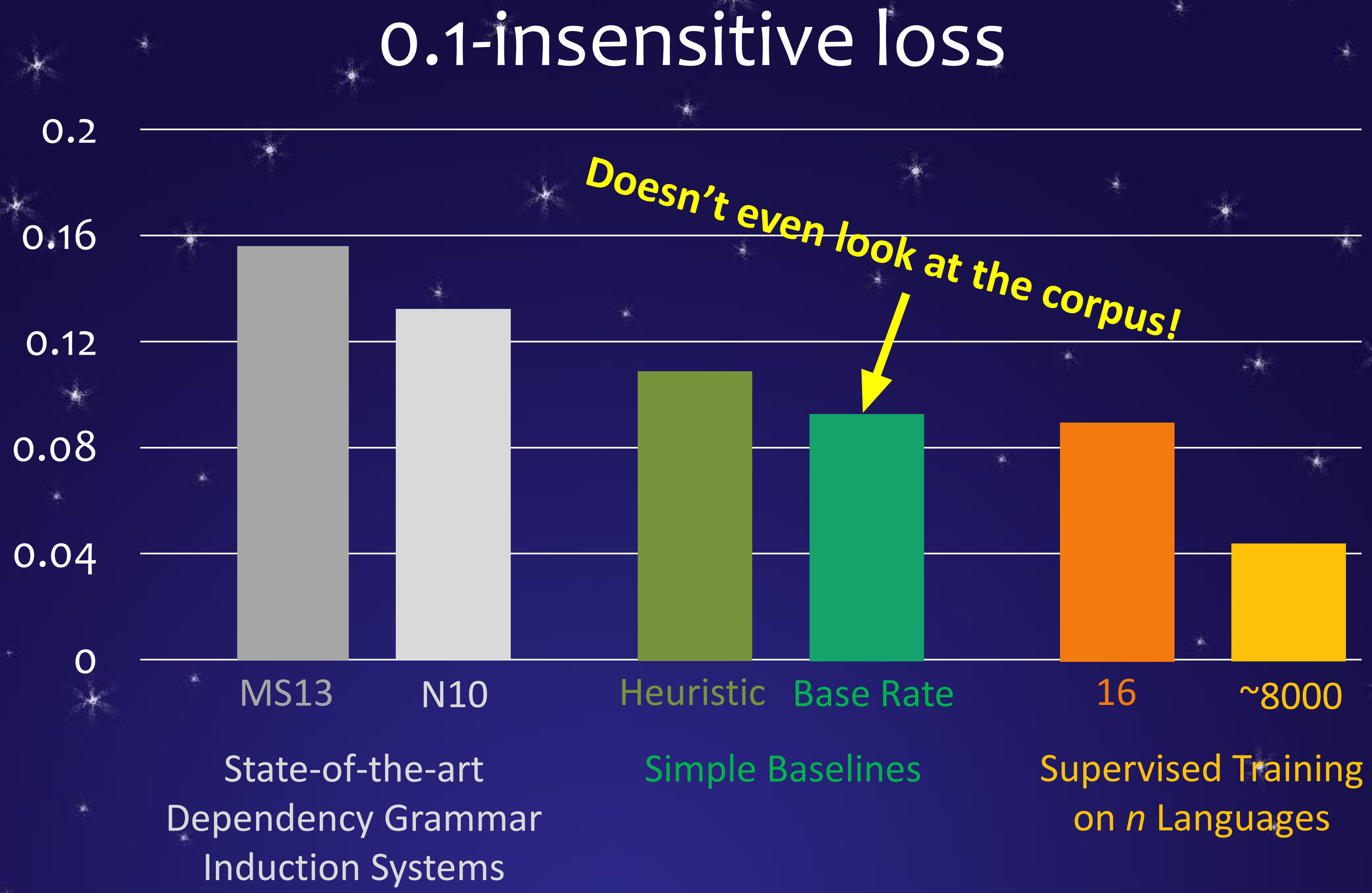
# Evaluation

- ε-insensitive loss

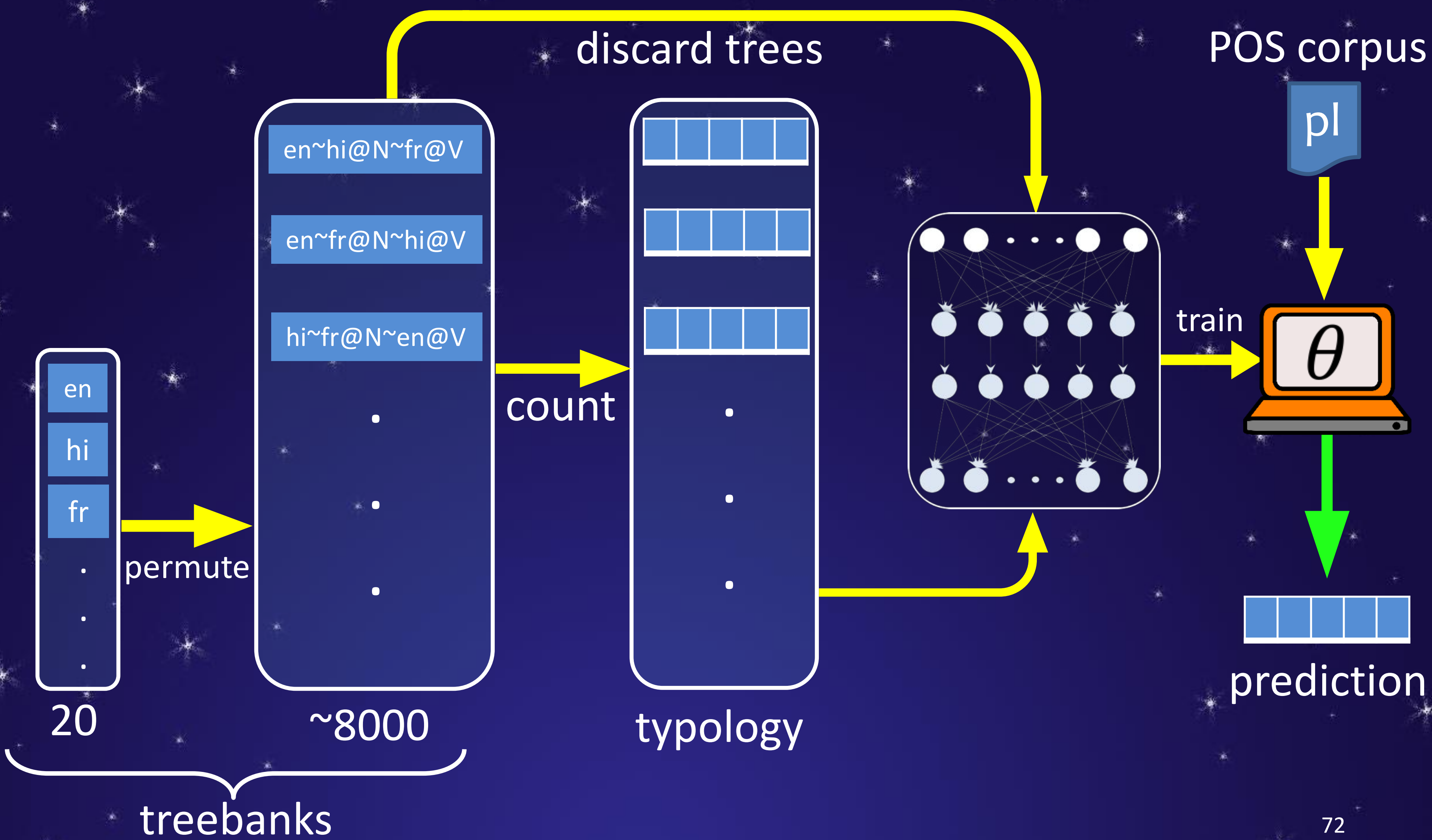


# Compared to Grammar Induction



# Summary: Training the System



# From Typology to Parsing

Corpus of POS-tags $\tilde{u}$

Galactic Dependencies

Learned mapper

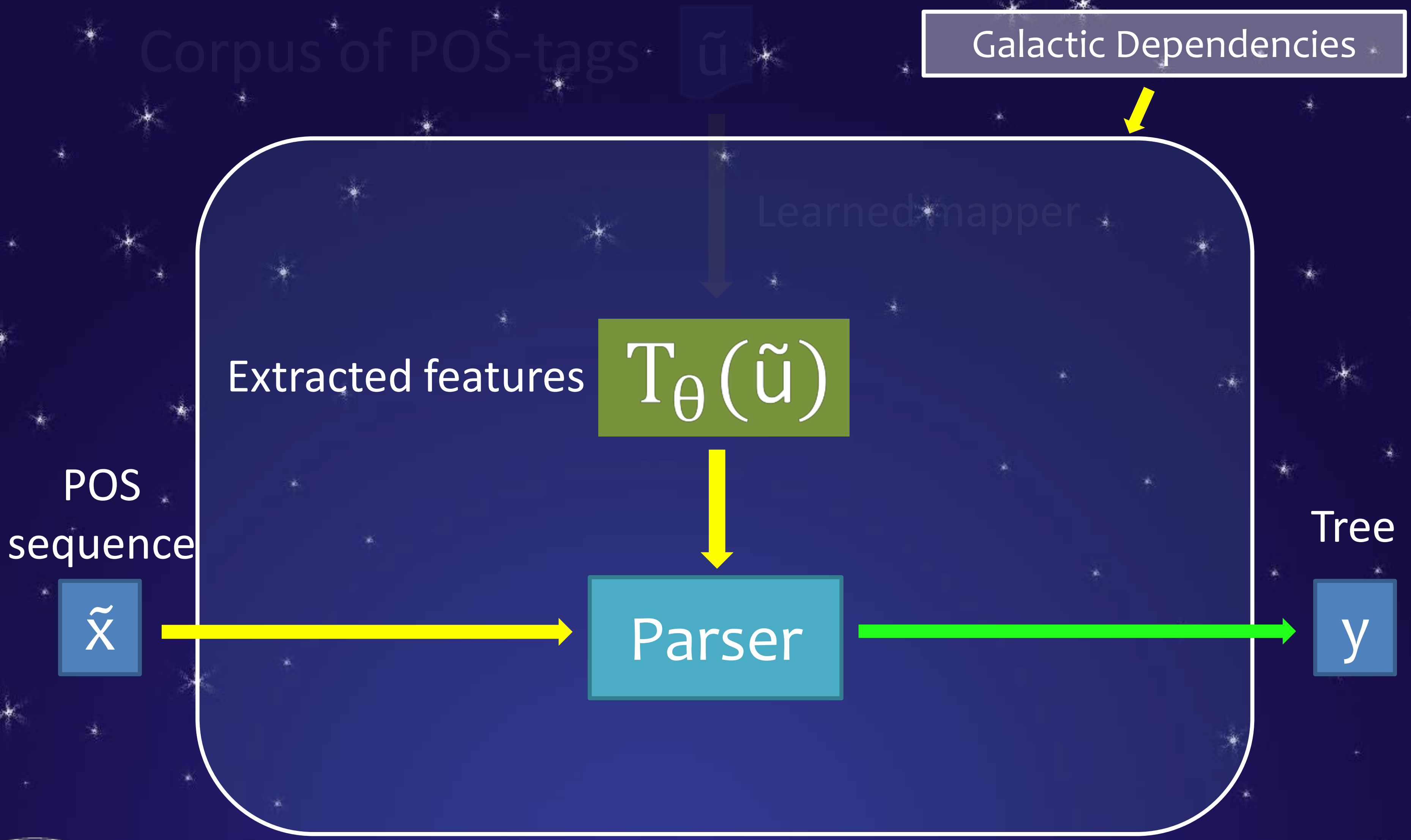
Extracted features $T_\theta(\tilde{u})$

POS sequence $\tilde{x}$

Parser

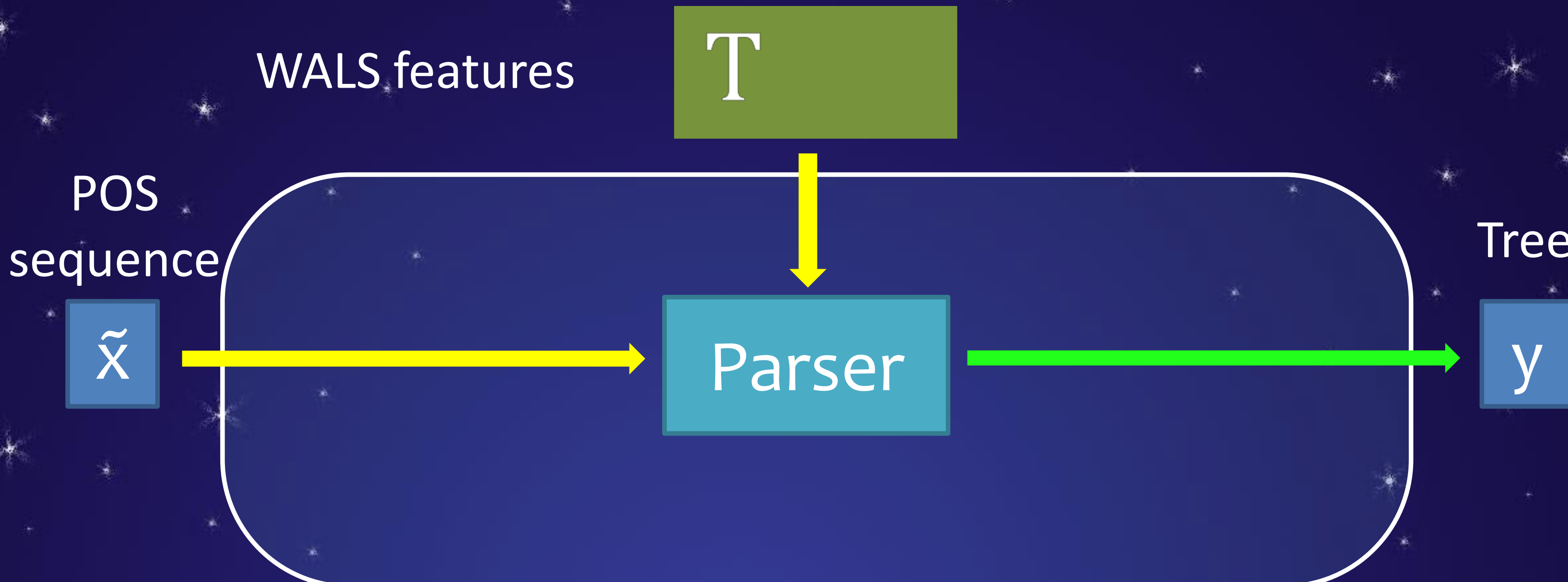
Tree $y$

# Our Typology or Standard Typology?

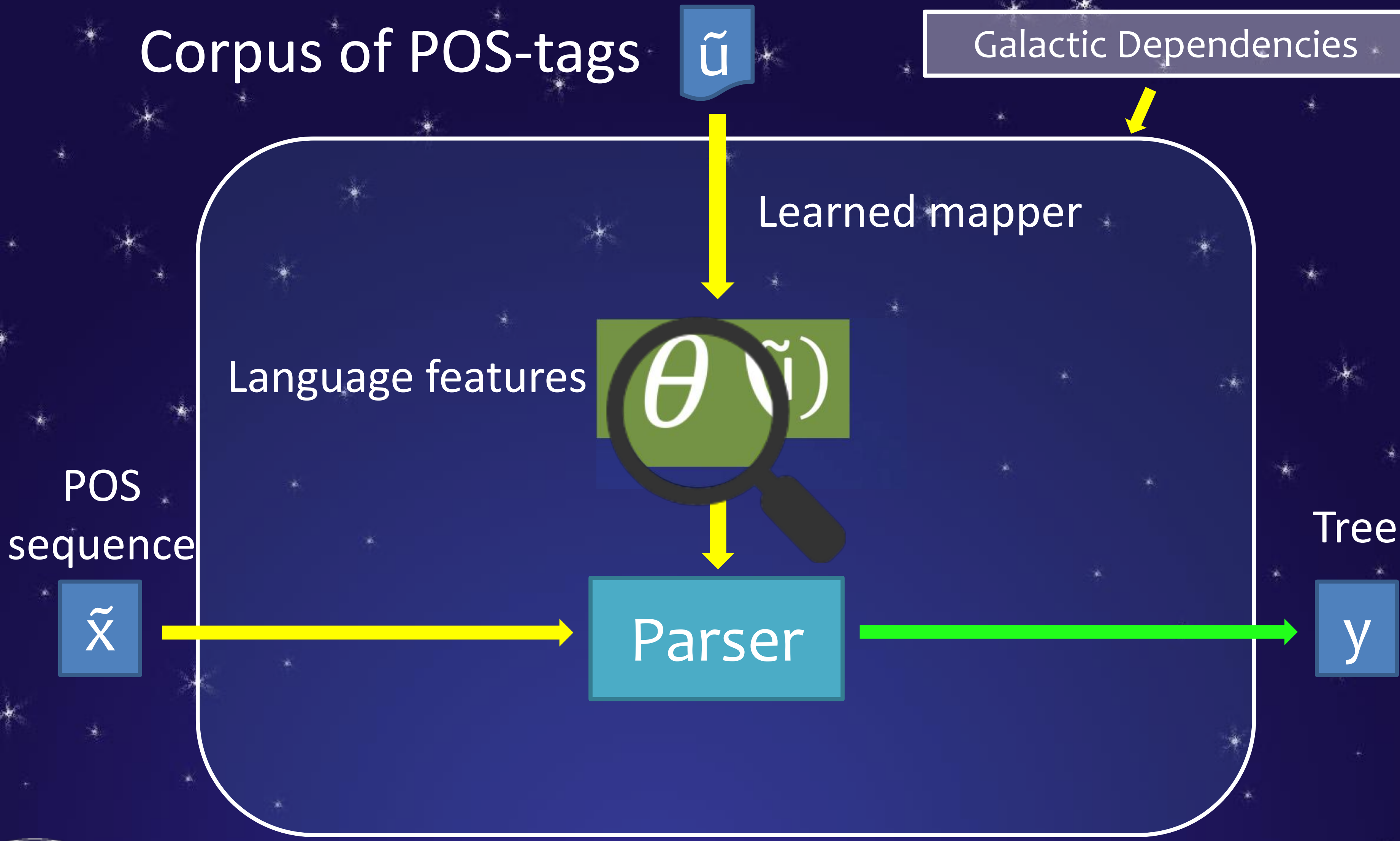


# Our Typology or Standard Typology?

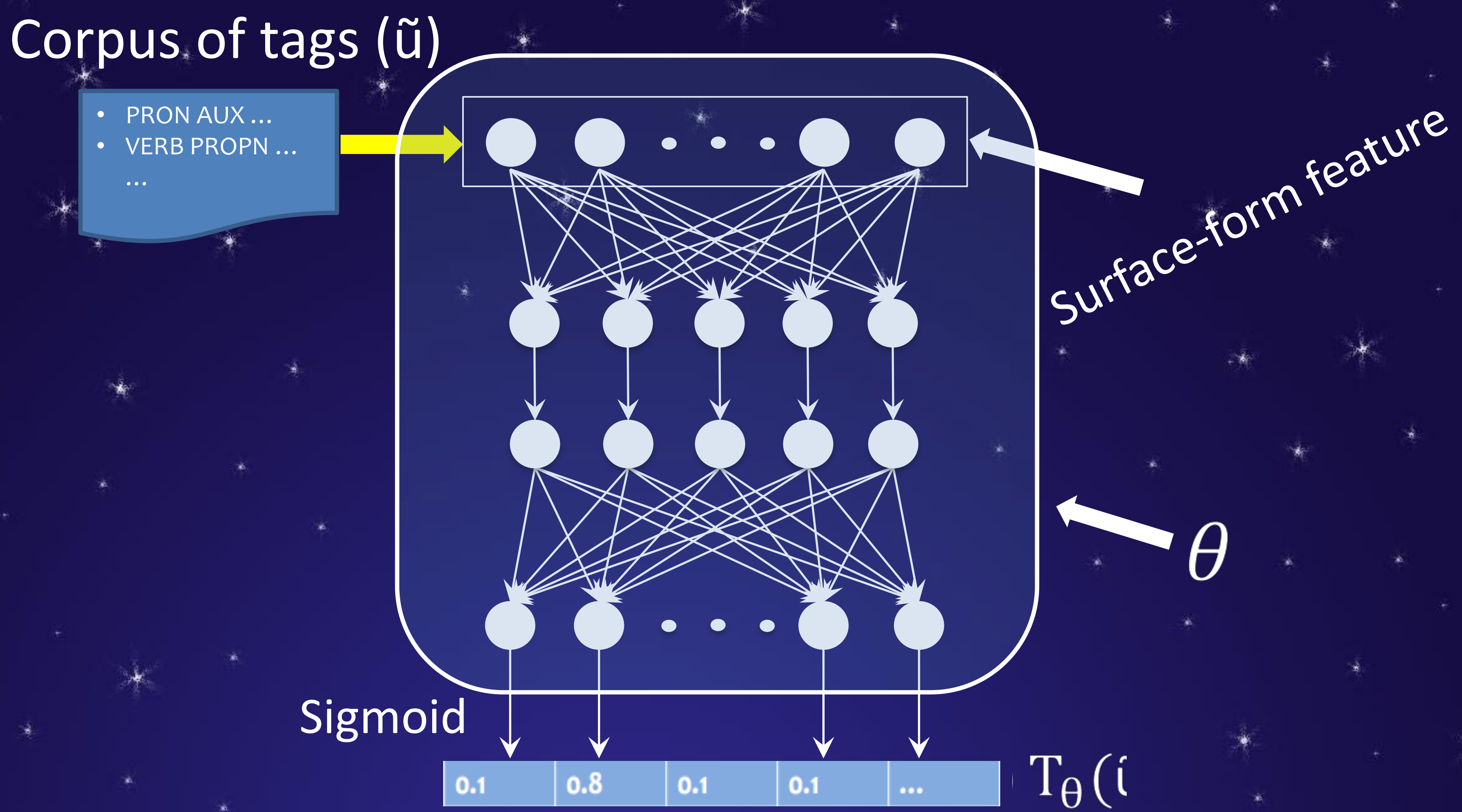
| ID | Feature Description | Values |
|---|---|---|
| 81A | Order of Subject, Object and Verb | SVO, SOV, VSO, VOS, OVS, OSV |
| 85A | Order of Adposition and Noun | Postpositions, Prepositions, Inpositions |
| 86A | Order of Genitive and Noun | Genitive-Noun, Noun-Genitive |
| 87A | Order of Adjective and Noun | Adjective-Noun, Noun-Adjective |
| 88A | Order of Demonstrative and Noun | Demonstrative-Noun, Noun-Demonstrative |
| 89A | Order of Numeral and Noun | Numeral-Noun, Noun-Numeral |



# Replacing the Typology Vector



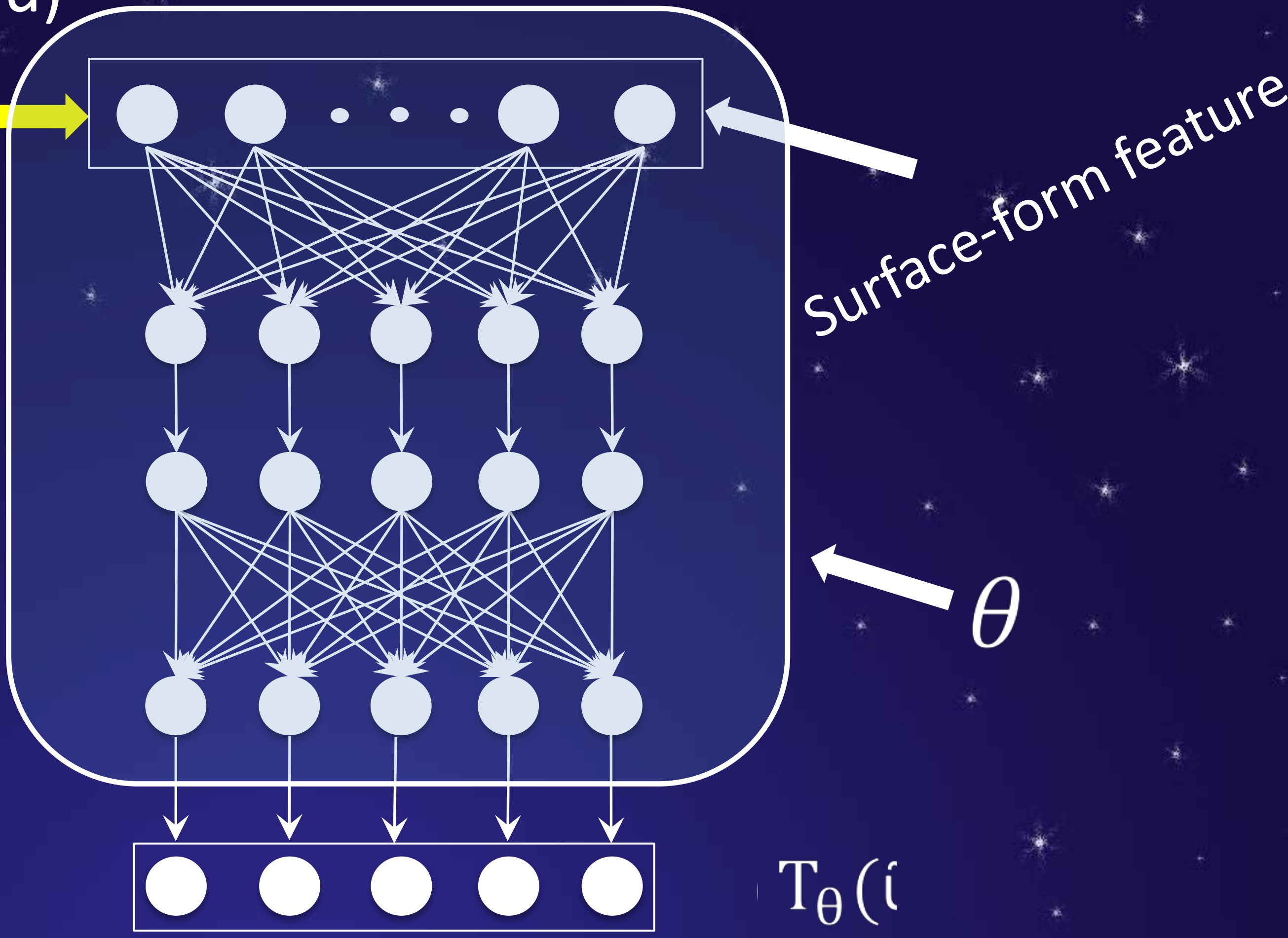
# Replacing the Typology Vector

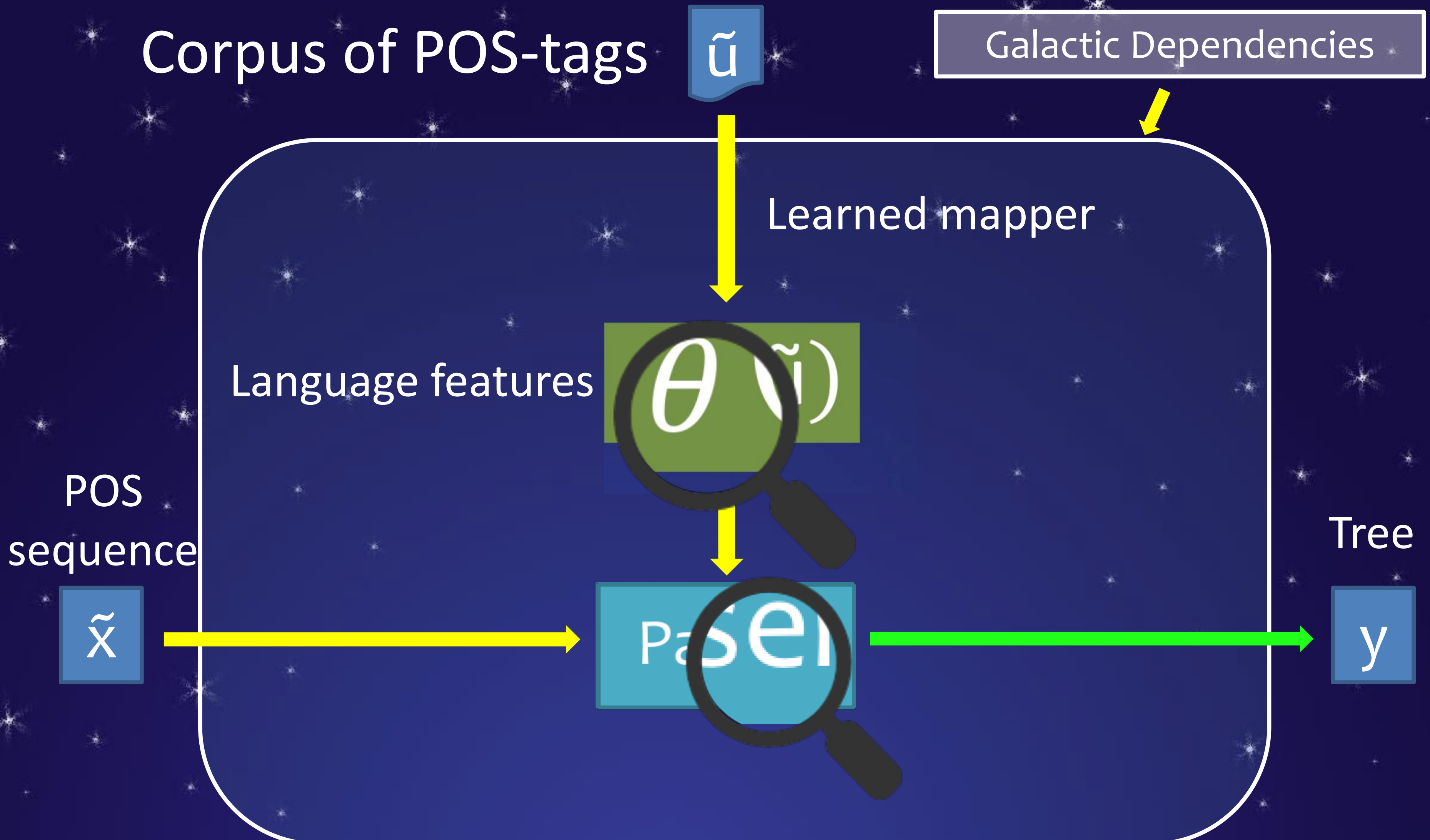


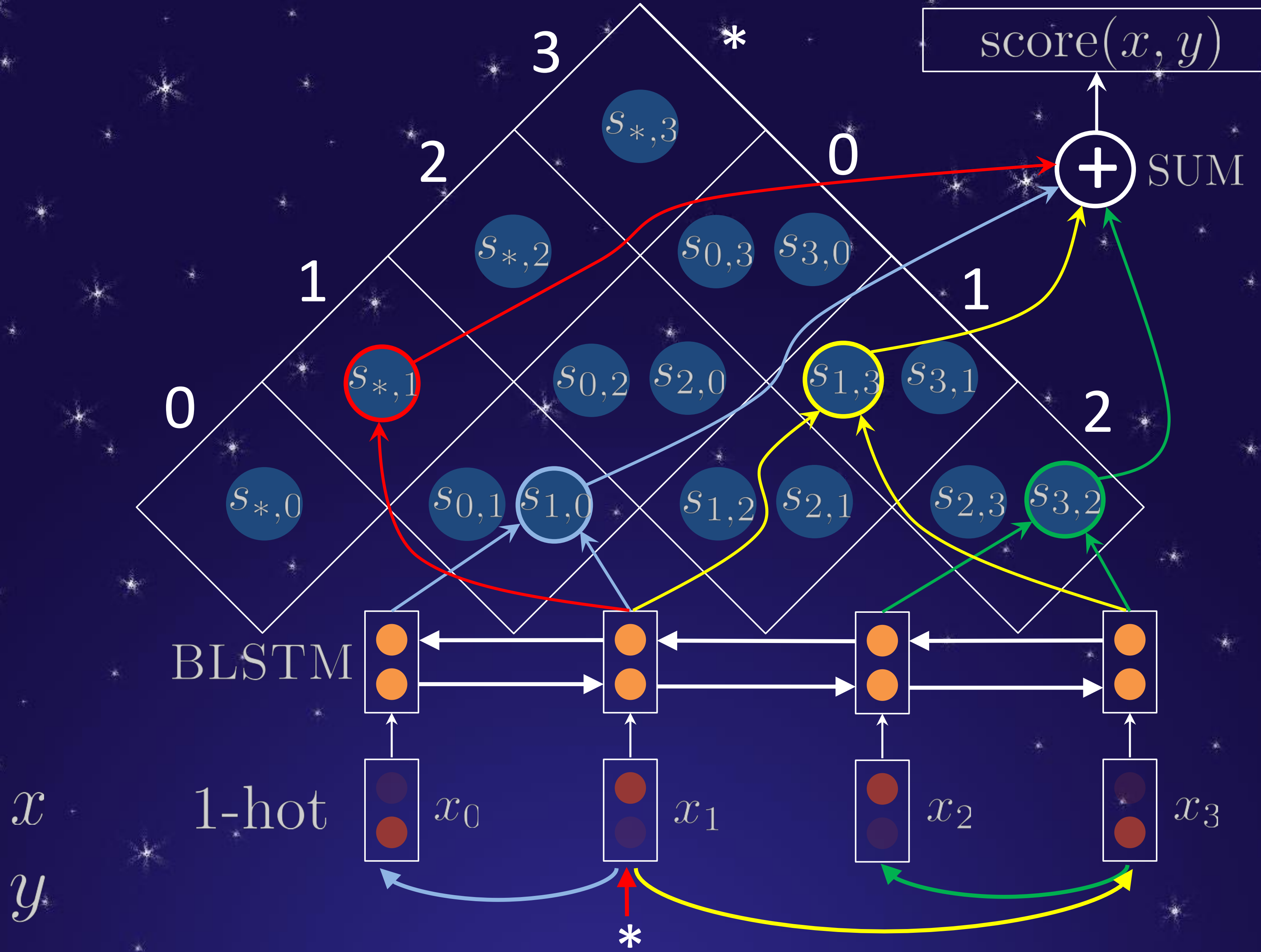
# Replacing the Typology Vector

Corpus of tags (ũ)

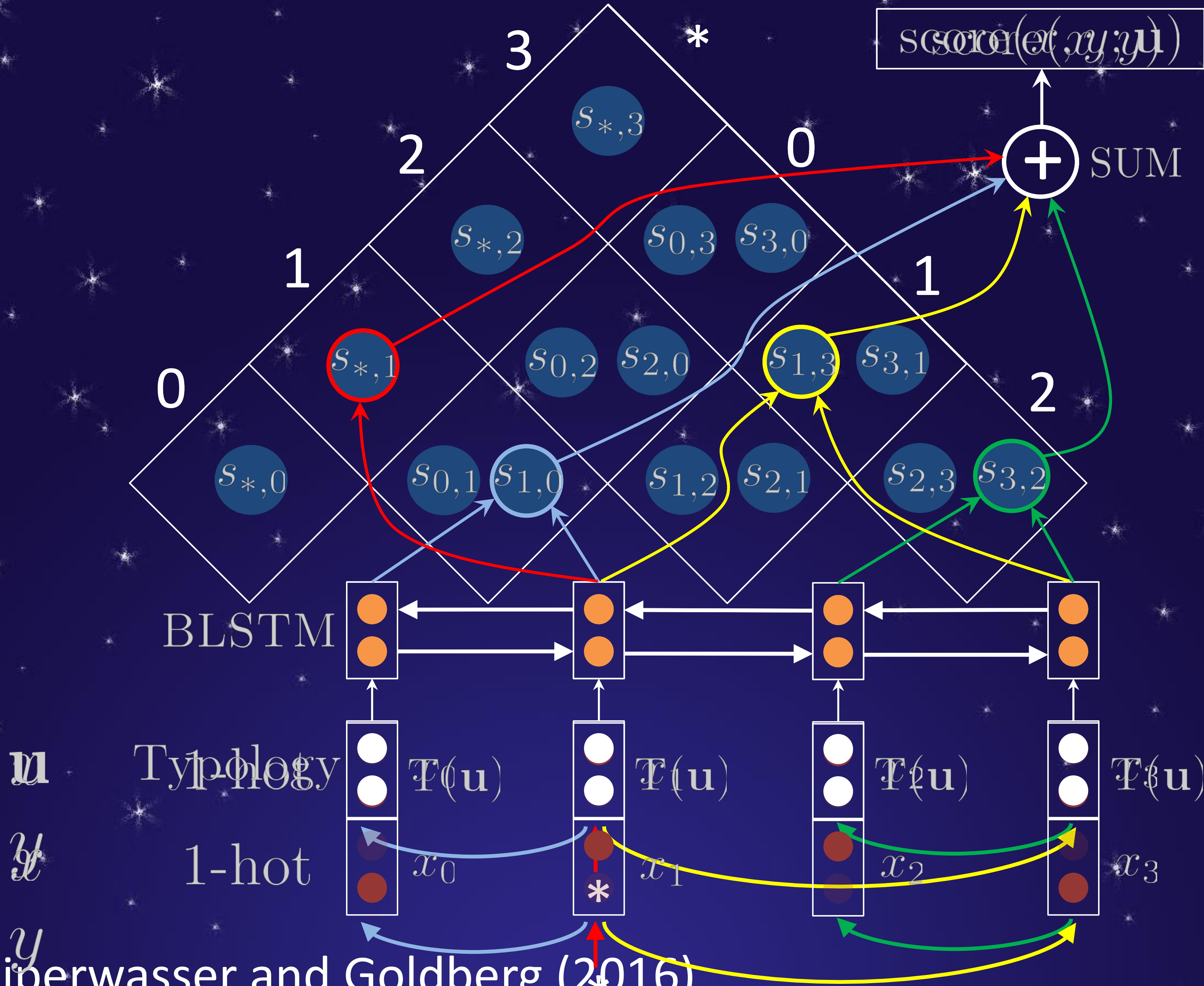
- PRON AUX …
- VERB PROPN …
- …




Corpus of POS-tags
ũ
Galactic Dependencies
Learned mapper
Language features
θ
POS sequence
x̃
Pa
Tree
y




Kiperwasser and Goldberg (2016)




Kiperwasser and Goldberg (2016)

# Supervised Training

POS-corpus

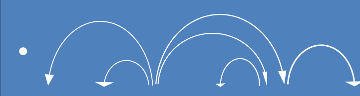
POS-treebank

ũ

(x̃, y)-pairs

Language 1

( • PRON AUX ...
• VERB PROPN ...
... ,



... )

Language 2

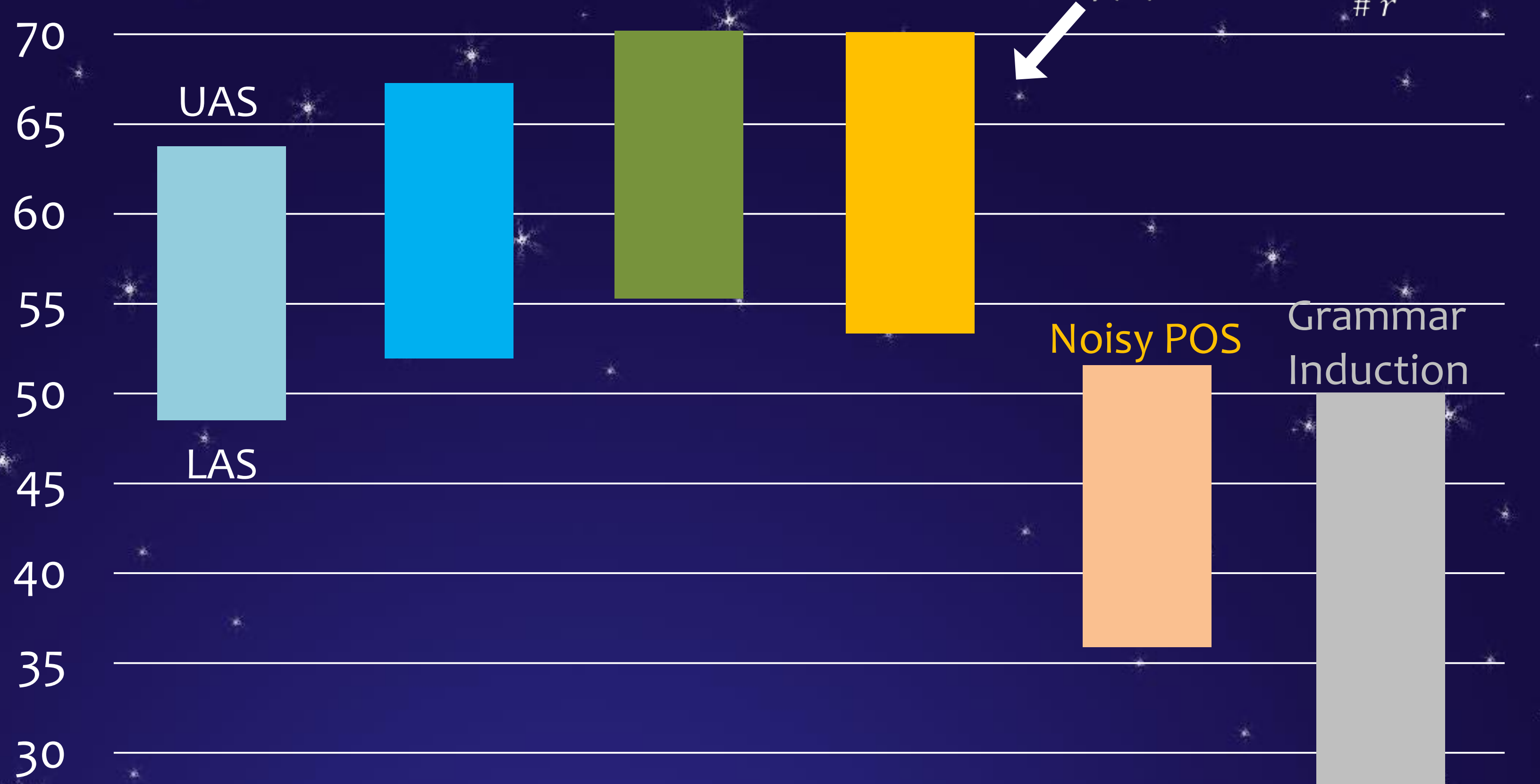
( • VERB NOUN...
• NOUN DET...
• NOUN ADJ ...
... ,

... )

.
.
.

# Avg. Parsing Performance

$$\text{Directionality}(r) = \frac{\#\, r \text{ from left to right}}{\#\, r}$$




| $T_\theta$ | ✗ | ✓ | ✓ | gold | ✓ |
|---|---|---|---|---|---|
| Syn. Lang. | ✗ | ✗ | ✓ | ✓ | ✓ |

# How can we recover linguists' structure?

Trust linguists' theory

Trust linguists' annotations

**Generative modeling**

$p(\theta)\, p(y \mid \theta)\, p(x \mid y, \theta)$

"Try to reason like a linguist"

(can figure out strange new languages)

**Conditional modeling**

$\hat{p}(x)\, p(y, \theta \mid x)$

"Mimic output of linguists"

(trained for accuracy on past languages)

# Future challenges

- Higher-quality synthetic languages
  - Non-projective orders, function morphemes, punctuation, restructuring, context-sensitive realization
  - Globally plausible realization systems
- Useful beyond the zero-shot scenario?
- Discover morphology and syntax jointly
- Handle raw text; exploit words and semantics
- Tasks beyond parsing

# Thanks!

JOHNS HOPKINS UNIVERSITY