Synthetic Data Made to Order: The Case of Parsing

Dingquan Wang and Jason Eisner
Department of Computer Science
Acknowledgement

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

Transfer Parsing
Transfer Parsing

we ate the tasty cake
Papa ate the caviar with a spoon
a blue cat ran away

Ma mère s'appelle Emilie Summer
Lundi, je retourne à l’école
C’est ma meilleure amie
J’aime beaucoup l’école
...

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

English Delex Treebank

French POS Corpus

train → parse

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
- ...

...
Delexicalized Transfer Parsing

train

parse

English’ Delex Treebank

French POS Corpus

NOUN VERB DET NOUN ADJ

NOUN VERB PART NOUN

DET NOUN ADJ VERB

PRON VERB ADP DET NOUN

...
Improve the surface similarity

French POS Corpus

NOUN VERB DET NOUN ADJ ADP NOUN
NOUN VERB PART NOUN
DET NOUN ADJ VERB
PRON VERB ADP DET NOUN
...

English'

English'

JOHNS HOPKINS UNIVERSITY
Improve the surface similarity

French POS Corpus

NOUN VERB DET NOUN ADJ
NOUN VERB PART NOUN
DET NOUN ADJ VERB
PRON VERB ADP ID
...

English

French
Improve the surface similarity

Source

Surface similarity

Transfer Parsing accuracy?

Target POS corpus
Single-Source Selection

Source languages

Target POS corpus

Surface similarity
Single-Source Selection

Source languages  Target POS corpus

POS-trigram similarity
Single-Source Selection

Source languages

Target POS corpus

train

parse

POS-trigram similarity
Synthetic Data

Source

Source'

Surface similarity

Target POS corpus

NOUN VERB DET NOUN ADJ
NOUN VERB PART NOUN
DET NOUN ADJ VERB
PRON VERB ADP DET NOUN
Synthetic Data

Source

Synthetic Source'

Surface similarity

Target POS corpus
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

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

Diagram:
- **det** → **amod** → **nsubj** → **dobj**
Surface Realization

Loglinear model

\[ p \left( \right) \]
How to find ?

Source’

Target POS corpus

Surface similarity
Scattershot

Source

random mutation!

Target
POS corpus

Surface similarity
This Paper: “Intelligent Design”

Source

Target

POS corpus

Bigram Distance

Back-Prop

0
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

\[
p \left( \begin{array}{c}
  \text{PRON} \\
  \text{nsubj} \\
  \text{ate} \\
  \text{dobj} \\
  \text{NOUN} \\
  \text{we} \\
  \text{cake}
\end{array} \right)
\]
Computing the Expected Counts by Dynamic Programming

\[ C_a(NOUN \; ADJ) \]
Computing the Expected Counts by Dynamic Programming

\[ C_a (\text{NOUN ADJ}) \]
Computing the Expected Counts by Dynamic Programming

\[ C_{a_2, a_0}(\text{NOUN ADJ}) \]

\[ C_{a_2}(\text{NOUN } \#) \times C_{a_0}(\# \text{ ADJ}) \]
Computing the Expected Counts by Dynamic Programming

\( C_\alpha(A) \)
Computing the Expected Counts by Dynamic Programming

$$C_a^\alpha(NOUN \ ADJ)$$

$$\sum_{a_0, a_1, a_2, a_3} C_{a_0, a_1, a_2, a_3}(NOUN \ ADJ)$$

$$C_{a_2, a_0}(NOUN \ ADJ) \quad C_{a_0, a_3}(NOUN \ ADJ) \quad C_{a_3, a_1}(NOUN \ ADJ)$$

$$C_{a_2}(NOUN \ ADJ) \quad C_{a_0}(NOUN \ ADJ) \quad C_{a_3}(NOUN \ ADJ) \quad C_{a_1}(NOUN \ ADJ)$$
Computing the Expected Counts by Dynamic Programming

\[ C_a(NOUN \ ADJ) \]

\[ C_{a_2,a_1}^2(NOUN \ ADJ) \]

\[ C_{a_0,a_3}^0(NOUN \ ADJ) \]

\[ C_{a_3,a_1}^3(NOUN \ ADJ) \]

\[ C_{a_1}(NOUN \ ADJ) \]
Computing the Expected Counts by Dynamic Programming

\[ C_\alpha(\text{NOUN ADJ}) = \sum_{\mathbf{p}(2,0,3,1 | \alpha)} p_{\mathbf{p}} \times C_\alpha^{(3,0,0,1)}(\text{NOUN ADJ}) \]

4! Permutations
Data

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

<table>
<thead>
<tr>
<th>Train</th>
<th>Test</th>
</tr>
</thead>
<tbody>
<tr>
<td>cs, es, fr, hi, de, it, la_itt, no, ar, pt, en, nl, da, fi, got, grc, et, la_proiel, grc_proiel, bg</td>
<td>la, hr, ga, he, hu, fa, ta, cu, el, ro, sl, ja_ktc, sv, fi_ftb, id, eu, pl</td>
</tr>
</tbody>
</table>
Is your method work?

Overall YES!

Averaged UAS over 376 pairs
Depends on the language family

Different family

Averaged UAS over 330 pairs

Source

Source'
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)

• Multi-sources!
  - One parser on many languages (Ammar et al. 2016, our poster tomorrow)
Wang and Eisner (2016)

Source 1  Source 2

Target
POS corpus

sexual
reproduction

Surface similarity
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

TACL 2016, TACL 2017, TACL 2018, EMNLP 2018
“What is a possible human language?”

• Typology in linguistics aims to describe the range of variation in human languages (Theories of Universal Grammar aim to explain it)

• If we know what languages can 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 many surface features; by examining many 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 Det Noun
Syntactic Typology (of English)

Subject-Verb-Object

nsubj  nsubj
N V V N

dobj  dobj
N V V N

nsubj  dobj
Papa ate a red apple at home
Syntactic Typology (of English)

Subject-Verb-Object

Prepositional

Adj-Noun

Subject: nsubj
Verb: nsubj
Object: nsubj

Preposition: case
Object: amod

Adjective: amod
Noun: amod

Example: Papa ate a red apple at home
Fine-grained Syntactic Typology (of English)

Subject-Verb-Object

Prepositional

Adj-Noun

nsubj

nsubj

case

case

amod

amod

✔

✘

✔

✘

✔

✘

✔

✘

✔

✘
Fine-grained Syntactic Typology (of English)

Subject-Verb-Object

Prepositional

Adj-Noun

nsubj

nssubj

case

case

amod

amod

N

V

V

V

V

N

N

N

N

ADP

ADP

A

A

0.96

0.04

0.96

0.04

0.97

0.03

0.04

0.96

0.04

0.96

0.04
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
- Postpositional
- Adj-Noun

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

Vector of length 57

| 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

<table>
<thead>
<tr>
<th>Language</th>
<th>nsubj</th>
<th>dobj</th>
<th>case</th>
<th>amod</th>
<th>...</th>
</tr>
</thead>
<tbody>
<tr>
<td>English</td>
<td>0.04</td>
<td>0.96</td>
<td>0.04</td>
<td>0.03</td>
<td>...</td>
</tr>
<tr>
<td>Japanese</td>
<td>0.0</td>
<td>0.0</td>
<td>1.0</td>
<td>0.0</td>
<td>...</td>
</tr>
<tr>
<td>Hindi</td>
<td>0.01</td>
<td>0.25</td>
<td>0.98</td>
<td>0.03</td>
<td>...</td>
</tr>
<tr>
<td>French</td>
<td>0.03</td>
<td>0.76</td>
<td>0.01</td>
<td>0.73</td>
<td>...</td>
</tr>
</tbody>
</table>
**Fine-grained Syntactic Typology**

Corpus of tags: ũ

<table>
<thead>
<tr>
<th>Tag Structure</th>
<th>Corpus Frequency</th>
</tr>
</thead>
<tbody>
<tr>
<td>NOUN VERB ADP NOUN PUNCT</td>
<td>0.04 0.96 0.04 0.03 ...</td>
</tr>
<tr>
<td>NOUN VERB PART NOUN PUNCT</td>
<td>0.0 0.0 1.0 0.0 ...</td>
</tr>
<tr>
<td>NOUN DET NOUN VERB PUNCT</td>
<td>0.01 0.25 0.98 0.03 ...</td>
</tr>
<tr>
<td>NOUN AUX NOUN ADP PUNCT</td>
<td>0.03 0.76 0.01 0.73 ...</td>
</tr>
<tr>
<td>NOUN NUM NOUN VERB</td>
<td></td>
</tr>
<tr>
<td>NOUN NOUN VERB PART</td>
<td></td>
</tr>
<tr>
<td>NOUN NOUN NOUN VERB</td>
<td></td>
</tr>
<tr>
<td>NOUN NOUN ADP PUNCT</td>
<td></td>
</tr>
</tbody>
</table>
Traditional approach: Grammar induction

- Yer/PRON amos/AUX yjja/VERB Ajjx/PROP 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/PROP „/PUNCT tat/PRON “/PUNCT yue/ADP han/NOUN ./PUNCT

S → NP VP 0.9
VP → VP PP 0.9
...
Grammar Induction

- Yer/PRON amos/AUX yjja/VERB Ajjx/PROP 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/PROP “/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)

- Yer/PRON amos/AUX yjja/VERB Ajx/PROP aat/ADP orrr/PRON ./PUNCT
- Per/NOUN ann/PRON se/NOUN in/PART wee/VERB ./PUNCT
- Con/VERB per/NOUN aat/ADP Ajx/PROP “/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 you 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?
Not holding out hope for a single trigger
But a combination of cues might work
Supervised learning

| Yer/PRON amos/AUX yjjia/VERB Ajjx/PROP Aat/ADP prrr/PRON ./PUNCT |
| Per/NOUN anmsee/NOUN in/P mywее/VERB ./PUNCT |
| Con/VERB per/NOUN aat/ADP Ajjx/PROP ./PUNCT tat/PRON “/PUNCT yue/ADP han/NOUN ./PUNCT |

(          ,       )

• You/PRON can/AUX ...
• Keep/VERB Google/PROP ...
• In/ADP my/PRON office/NOUN ...

(          ,       )

You/PRON can/AUX ...
Keep/VERB Google/PROP ...
In/ADP my/PRON office/NOUN ...

(          ,       )

• /PRON /AUX ...
• /VERB /PROP ...
• /ADP /PRON /NOUN ...

0.25 0.8 1.0 ...
0.04 0.96 0.04 ...
0.03 0.76 0.01 ...

training data
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?

• You/Pron can/Aux ...
• Keep/Verb Google/Propn ...
• In/Adp my/Pron office/Noun ...

• 0.04 0.96 0.04 ...
• 0.03 0.76 0.01 ...
• 0.05 0.45 0.50 ...
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,…. 

(6, 6)   (6, 6)   (6, 6)
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)

Hindi — Superstrate

English — Substrate

verb order

Japanese — Superstrate

noun order
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) $\rightarrow$ SOV (Hindi)

Language: English

You must feel the Force around you.
Permute the children of verbs

• SVO (English) $\rightarrow$ 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 $3^3 = 50,000$ extraterrestrial treebanks
  – "Galactic Dependencies" treebanks
  – Still in Universal Dependencies Format

You feel you must feel around the Force.
How exactly do we permute?

You must feel Force you.
You must feel Force you.
Sampling

How many possible orders?

5!
\( p( \cdot | SVO, \text{Hindi}) \)

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>S O V</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>S V O</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>O S V</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>O V S</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>V S O</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>V O S</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

S:nsubj  
O:dobj  
V:VERB  
BOS: Beginning of sentence
Each order has features as shown
- Train a log-linear model on Hindi treebank
- Sample from it to reorder English trees

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>SOV</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>SVO</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>OSV</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>OVS</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>VSO</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>VOS</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

S:nsubj
O:dobj
V:VERB

BOS: Beginning of sentence
\[ \mathbf{p}(\cdot | \text{SO VO}, \text{Hindi}) \]

<table>
<thead>
<tr>
<th>Order</th>
<th>Prob.</th>
<th>BOS S adj.</th>
<th>S(&lt;V</th>
<th>O(&lt;V</th>
<th>SO adj.</th>
</tr>
</thead>
<tbody>
<tr>
<td>S O V</td>
<td>0.8</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>S V O</td>
<td>0.03</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>O S V</td>
<td>0.1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>O V S</td>
<td>0.02</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>V S O</td>
<td>0.03</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>V O S</td>
<td>0.02</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>
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
Evaluation: Parsability

- real
- synthetic
Evaluation: Diversity

distance ≈ parsing transfer accuracy
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 $\sin(x) \approx y$, and interpolate.
  - In ML: This work! “Scattershot” precomputation of many reasonable observed sentences, and then learn how to work backwards from $y$ to $x$ …
Parser

Language features

POS sequence

Tree

Galactic Dependencies

Corpus of POS-tags

$\tilde{\mu}$

$T_\theta(\tilde{\mu})$

Learned mapper

POS sequence $\tilde{X}$

$y$
Corpus of POS-tags \( \tilde{X} \) \( \rightarrow \) Language features \( T_\theta(\tilde{u}) \) \( \rightarrow \) Learned mapper \( \tilde{u} \) \( \rightarrow \) Galactic Dependencies
Wang and Eisner (2017)

Corpus of POS-tags → \( T_\theta (\tilde{\nu}) \)

Language features

Fine-grained syntactic typology

Learned mapper

Galactic Dependencies
Prediction of Syntactic Typology

Corpus of POS-tags \( \tilde{\mu} \)

Galactic Dependencies

Learned mapper

Language features \( T_\theta(\tilde{\mu}) \)
Prediction of Syntactic Typology

Corpus of POS-tags → Learned mapper → \( T_\theta(\tilde{u}) \) → Galactic Dependencies

Language features

Trained with supervision!
Supervised Training

POS-corpus

Language 1

• PRON AUX ...
• VERB PROPN ...

Language 2

• VERB NOUN...
• NOUN DET...
• NOUN ADJ ...

Vector of length 57

True Typology

Directionality(r) = \frac{\# r from left to right}{\# r}
Prediction of Syntactic Typology

Corpus of POS-tags

Learned mapper

Language features

\( \theta(\tilde{u}) \)

Galactic Dependencies
Hand-designed features

Corpus of tags ($\tilde{\eta}$)

- PRON
- AUX
- VERB
- PROPN
- ...

$T_\theta(i)$

Local-window POS counts and ratios

Sigmoid

$\theta$
LSTM features

max-pooling,
(also softmax-pooling by varying $\beta$, 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

<table>
<thead>
<tr>
<th>Train</th>
<th>Test</th>
</tr>
</thead>
<tbody>
<tr>
<td>cs, es, fr, hi, de, it, la itt, no, ar, pt en, nl, da, fi, got, grc, et, la proiel, grc proiel, bg</td>
<td>hr, ga, he, hu, fa, ta, cu, el, ro, sl, ja ktc, sv, id, eu, pl</td>
</tr>
</tbody>
</table>
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

• $\epsilon$-insensitive loss
Compared to Grammar Induction

0.1-insensitive loss

State-of-the-art Dependency Grammar Induction Systems

Simple Baselines

Supervised Training on $n$ Languages

Doesn’t even look at the corpus!
Summary: Training the System

POS corpus

pl

train

θ

discard trees

count

typology

treebanks

~8000

en~hi@N~fr@V

en~fr@N~hi@V

hi~fr@N~en@V

permute

20
From Typology to Parsing

Corpus of POS-tags \(\tilde{X}\) \(\xrightarrow{T_\theta(\tilde{\mu})}\) \(\tilde{\mu}\) \(\xrightarrow{\text{Learned mapper}}\) \(\text{Galactic Dependencies}\)

Extracted features

POS sequence \(\tilde{X}\)

Parser

Tree
Our Typology or Standard Typology?

Corpus of POS-tags → \( T_\theta (\tilde{u}) \) → Galactic Dependencies

Extracted features → Parser → Tree

POS sequence → \( \tilde{x} \) → \( x \) → \( y \)
**Our Typology or Standard Typology?**

<table>
<thead>
<tr>
<th>ID</th>
<th>Feature Description</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>81A</td>
<td>Order of Subject, Object and Verb</td>
<td>SVO, SOV, VSO, VOS, OVS, OVS, OSV</td>
</tr>
<tr>
<td>85A</td>
<td>Order of Adposition and Noun</td>
<td>Postpositions, Prepositions, Inpositions</td>
</tr>
<tr>
<td>86A</td>
<td>Order of Genitive and Noun</td>
<td>Genitive-Noun, Noun-Genitive</td>
</tr>
<tr>
<td>87A</td>
<td>Order of Adjective and Noun</td>
<td>Adjective-Noun, Noun-Adjective</td>
</tr>
<tr>
<td>88A</td>
<td>Order of Demonstrative and Noun</td>
<td>Demonstrative-Noun, Noun-Demonstrative</td>
</tr>
<tr>
<td>89A</td>
<td>Order of Numeral and Noun</td>
<td>Numeral-Noun, Noun-Numeral</td>
</tr>
</tbody>
</table>

**Diagram:**
- **POS sequence**: $\tilde{X}$
- **WALS features**: T
- **Parser**
- **Tree**: y
Replacing the Typology Vector

Corpus of POS-tags → \(\tilde{\mathbf{u}}\) → Galactic Dependencies

\(\tilde{\mathbf{X}}\) → Language features → \(\theta(\tilde{\mathbf{u}})\) → \(\tilde{\mathbf{X}}\) → Parser → \(y\) → Tree
Replacing the Typology Vector

Corpus of tags (û)

• PRON AUX...
• VERB PROPN...
• ...

Sigmoid

θ

\[ T_\theta (\hat{\text{}}) \]
Replacing the Typology Vector

Corpus of tags ($\hat{u}$)

- PRON AUX ...
- VERB PROPN ...

$T_{\theta}(i)$

Surface-form feature

$\theta$
Corpus of POS-tags → Learned mapper → Galactic Dependencies

POS sequence → Language features → Parser → Tree
Kiperwasser and Goldberg (2016)
Supervised Training

POS-corpus

Language 1

Language 2

POS-treebank

(\(\tilde{x}, y\))-pairs
Avg. Parsing Performance

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

<table>
<thead>
<tr>
<th>T_0</th>
<th>Syn. Lang.</th>
</tr>
</thead>
<tbody>
<tr>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>gold</td>
<td>✔</td>
</tr>
<tr>
<td>✔</td>
<td>✔</td>
</tr>
</tbody>
</table>
How can we recover linguists’ structure?

Trust linguists’ theory

Generative modeling
\[ p(\theta) p(y | \theta) p(x | y, \theta) \]

“Try to reason like a linguist”
(can figure out strange new languages)

Trust linguists’ annotations

Conditional modeling
\[ \hat{p}(x) p(y, \theta | x) \]

“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!