Zero-resource Dependency Parsing: Boosting Delexicalized Cross-lingual Transfer with Linguistic Knowledge

Lauriane Aufrant, Guillaume Wisniewski, François Yvon

December 13, 2016

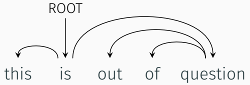
COLING'16

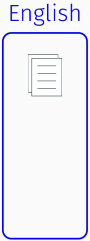
Zero-resource scenario:

$$\left.\begin{cases}\text{No annotated data}\\ \text{No parallel data}\\ \text{No raw data}\end{cases}\right. \Longrightarrow \text{delexicalized cross-lingual transfer}$$

Zero-resource scenario:

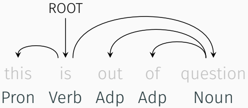
$$\left\{\begin{array}{l}\text{No annotated data}\\ \text{No parallel data}\\ \text{No raw data}\end{array}\right. \Longrightarrow \text{delexicalized cross-lingual transfer}$$

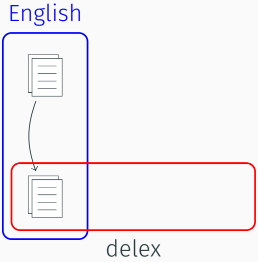




Zero-resource scenario:

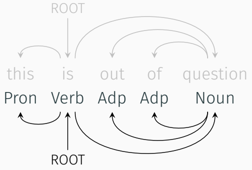
$$\begin{cases}\text{No annotated data}\\ \text{No parallel data}\\ \text{No raw data}\end{cases} \implies \text{delexicalized cross-lingual transfer}$$

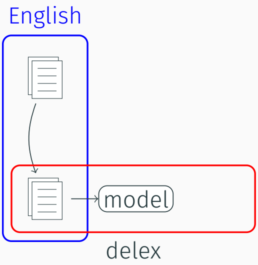




Zero-resource scenario:

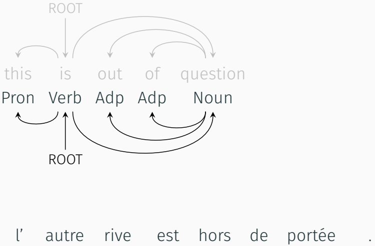
$$\left\{\begin{array}{l}\text{No annotated data}\\ \text{No parallel data}\\ \text{No raw data}\end{array}\right. \Longrightarrow \text{delexicalized cross-lingual transfer}$$

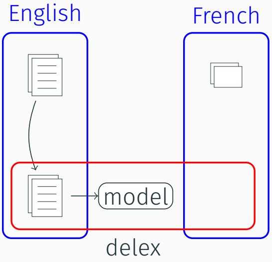




Zero-resource scenario:

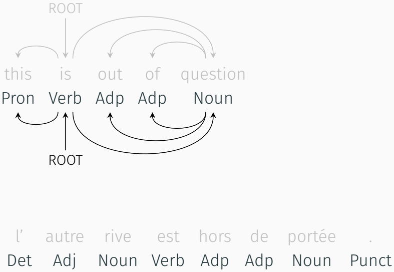
$$\left\{\begin{array}{l}\text{No annotated data}\\ \text{No parallel data}\\ \text{No raw data}\end{array}\right. \Longrightarrow \text{delexicalized cross-lingual transfer}$$

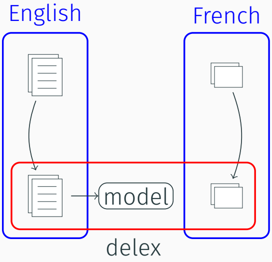




Zero-resource scenario:

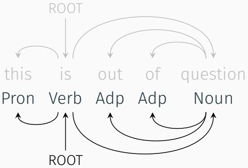
$$\left\{\begin{array}{l}\text{No annotated data}\\ \text{No parallel data}\\ \text{No raw data}\end{array}\right. \Longrightarrow \text{delexicalized cross-lingual transfer}$$

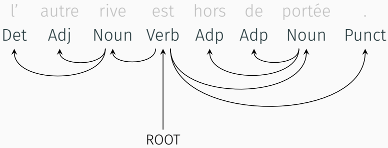


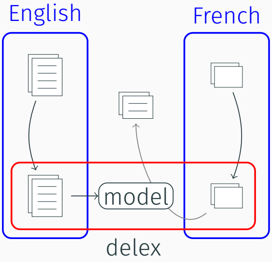


Zero-resource scenario:

$$\begin{cases} \text{No annotated data} \\ \text{No parallel data} \\ \text{No raw data} \end{cases} \Longrightarrow \text{delexicalized cross-lingual transfer}$$

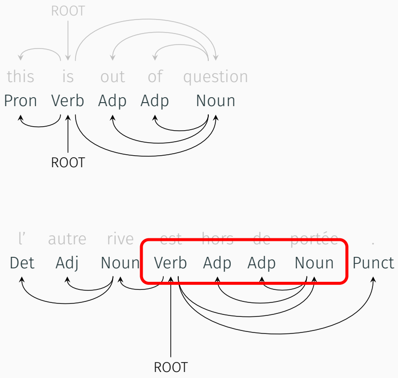


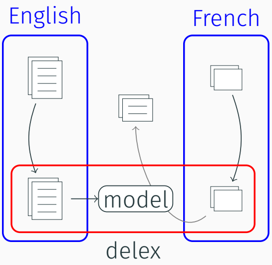




Zero-resource scenario:

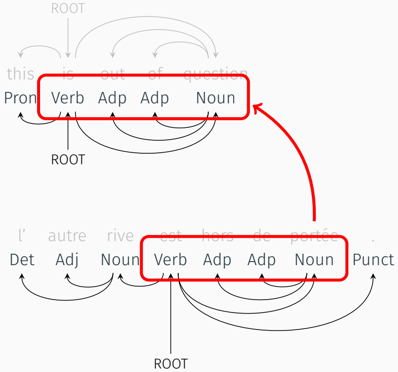
$$\begin{cases}\text{No annotated data}\\ \text{No parallel data}\\ \text{No raw data}\end{cases} \Longrightarrow \text{delexicalized cross-lingual transfer}$$

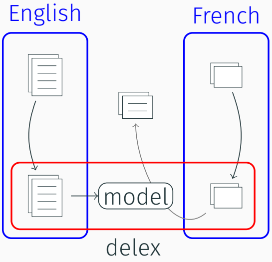




Zero-resource scenario:

$$\left\{\begin{array}{l}\text{No annotated data}\\ \text{No parallel data}\\ \text{No raw data}\end{array}\right. \Longrightarrow \text{delexicalized cross-lingual transfer}$$





An adjective close to a noun depends on this noun.

An adjective close to a noun depends on this noun.

An adjective close to a noun depends on this noun.

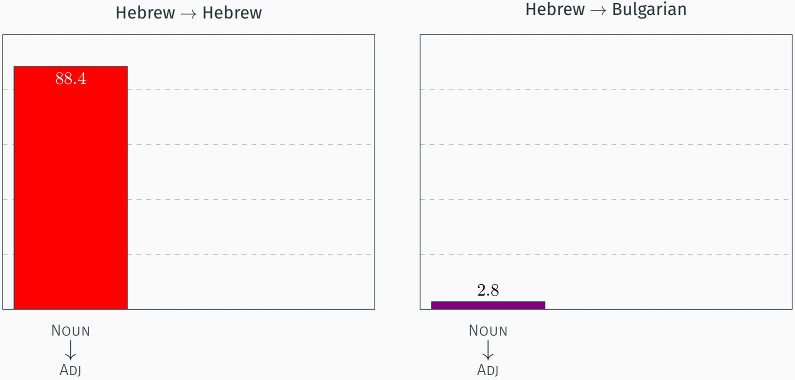
True in...

- ✓ English
- ✓ French
- ✓ Hebrew
- ✓ Bulgarian

An adjective close to a noun depends on this noun.

True in...

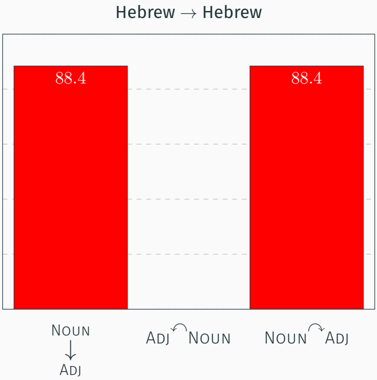
- ✓ English
- ✓ French
- ✓ Hebrew
- ✓ Bulgarian

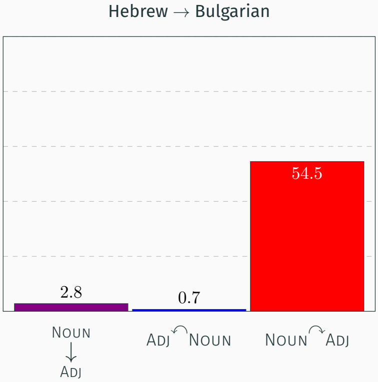


An adjective close to a noun depends on this noun.

True in...

- ✓ English
- ✓ French
- ✓ Hebrew
- ✓ Bulgarian





A step towards scaling to the 7,000 languages in the world

↪ zero-resource dependency transfer

Many transfer errors are easy to avoid

↪ regular divergences between source and target

↪ word order issues

Our approach: leveraging previous works in linguistics (WALS)

↪ +3 UAS on average

↪ very efficient on some error types: up to +90 UAS

A fine-grained analysis across various language pairs

↪ 6,000+ experiments

Outline

1. Transition-based dependency parsing
2. Impact of word order on transfer
3. Leveraging WALS
4. Leveraging raw data
5. Wrap-up

Transition-based dependency parsing

Parsing with the ArcEager system

$\perp$ | They ate okonomiyaki at Ōsaka

stack buffer

They ate okonomiyaki at Ōsaka

Classifier

Parsing with the ArcEager system

They | ate okonomiyaki at Ōsaka

stack buffer

They ate okonomiyaki at Ōsaka

Classifier

Shift

Parsing with the ArcEager system

They | ate okonomiyaki at Ōsaka

stack buffer

$\uparrow$ $\uparrow$ $\uparrow$ $\uparrow$

s_1 s_0 n_0 n_1

s_1 = $\varnothing$

s_0 = They

n_0 = ate

n_1 = okonomiyaki

$\downarrow$

Classifier

They ate okonomiyaki at Ōsaka

Parsing with the ArcEager system

They | ate okonomiyaki at Ōsaka

stack buffer

s_1 s_0 n_0 n_1

s_1 = ∅
s_0 = THEY
n_0 = ATE
n_1 = OKONOMIYAKI

CLASSIFIER

LEFT

They ate okonomiyaki at Ōsaka

Parsing with the ArcEager system

$\perp$ | ate okonomiyaki at Ōsaka

stack buffer

They ate okonomiyaki at Ōsaka

Classifier

Left

Parsing with the ArcEager system

$\perp$ | ate okonomiyaki at Ōsaka

stack buffer

They ate okonomiyaki at Ōsaka

Classifier

Parsing with the ArcEager system

$\perp$ | ate okonomiyaki at Ōsaka

stack buffer

They ate okonomiyaki at Ōsaka

CLASSIFIER

SHIFT

Parsing with the ArcEager system

ate | okonomiyaki at Ōsaka

stack buffer

They ate okonomiyaki at Ōsaka

CLASSIFIER

SHIFT

Parsing with the ArcEager system

ate | okonomiyaki at Ōsaka

stack buffer

s_1 s_0 n_0 n_1

$$
\begin{aligned}
s_1 &= \varnothing \\
s_0 &= \text{ATE} \\
n_0 &= \text{OKONOMIYAKI} \\
n_1 &= \text{AT}
\end{aligned}
$$

CLASSIFIER

They ate okonomiyaki at Ōsaka

Parsing with the ArcEager system

ate | okonomiyaki at Ōsaka

stack buffer

s_1 s_0 n_0 n_1

s_1	=	∅
s_0	=	ATE
n_0	=	OKONOMIYAKI
n_1	=	AT

They ate okonomiyaki at Ōsaka

CLASSIFIER

RIGHT

Parsing with the ArcEager system

ate okonomiyaki | at Ōsaka

stack buffer

They ate okonomiyaki at Ōsaka

CLASSIFIER

RIGHT

Parsing with the ArcEager system

ate okonomiyaki | at Ōsaka

stack buffer

s_1 s_0 n_0 n_1

s_1 = ATE
s_0 = OKONOMIYAKI
n_0 = AT
n_1 = ŌSAKA

CLASSIFIER

They ate okonomiyaki at Ōsaka

Parsing with the ArcEager system

ate okonomiyaki | at Ōsaka

stack buffer

s_1 s_0 n_0 n_1

They ate okonomiyaki at Ōsaka

s_1 = ATE
s_0 = OKONOMIYAKI
n_0 = AT
n_1 = ŌSAKA

CLASSIFIER

REDUCE

Parsing with the ArcEager system

ate | at Ōsaka

stack buffer

They ate okonomiyaki at Ōsaka

CLASSIFIER

REDUCE

Parsing with the ArcEager system

ate | at Ōsaka

stack buffer

s_1 s_0 n_0 n_1

s_1 = $\varnothing$
s_0 = ATE
n_0 = AT
n_1 = ŌSAKA

CLASSIFIER

They ate okonomiyaki at Ōsaka

Parsing with the ArcEager system

$\boxed{\text{ate}}$ | at Ōsaka

stack buffer

s_1 s_0 n_0 n_1

They ate okonomiyaki at Ōsaka

s_1 = $\varnothing$

s_0 = ATE

n_0 = AT

n_1 = ŌSAKA

↓

CLASSIFIER

↓

SHIFT

Parsing with the ArcEager system

ate at | Ōsaka

stack buffer

They ate okonomiyaki at Ōsaka

Classifier

Shift

Parsing with the ArcEager system

ate at | Ōsaka

stack buffer

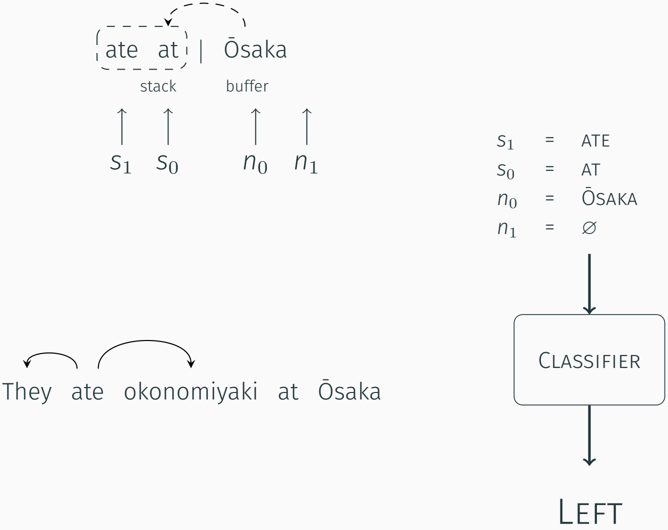
s_1 s_0 n_0 n_1

s_1 = ATE
s_0 = AT
n_0 = ŌSAKA
n_1 = $\varnothing$

CLASSIFIER

They ate okonomiyaki at Ōsaka

Parsing with the ArcEager system



Parsing with the ArcEager system

ate | Ōsaka

stack buffer

They ate okonomiyaki at Ōsaka

CLASSIFIER

LEFT

Parsing with the ArcEager system

ate | Ōsaka

stack buffer

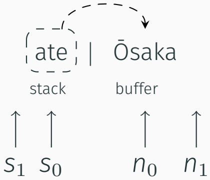
s_1 s_0 n_0 n_1

s_1 = $\varnothing$
s_0 = ATE
n_0 = ŌSAKA
n_1 = $\varnothing$

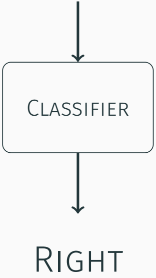
CLASSIFIER

They ate okonomiyaki at Ōsaka

Parsing with the ArcEager system



$s_1 = \varnothing$
$s_0 =$ ATE
$n_0 =$ ŌSAKA
$n_1 = \varnothing$



Parsing with the ArcEager system

Classifier

Right

Parsing with the ArcEager system

Parsing with the ArcEager system

ate Ōsaka | ⊥

stack buffer

Classifier

Reduce

Parsing with the ArcEager system

ate | $\perp$

stack buffer

Classifier

Reduce

Parsing with the ArcEager system

ate | $\perp$

stack buffer

Classifier

Parsing with the ArcEager system

ate | ⊥

stack buffer

Classifier

Impact of word order on transfer

Towards language-independent data representations

- **Words:** delexicalized, common tagset
  ✓ UPOS [Petrov et al., 2012]

- **Dependencies:** common guidelines
  ✓ Universal Dependencies [Nivre et al., 2016]

- … other regular divergences?

Impact of word order

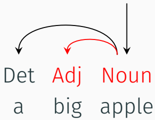
At data level:

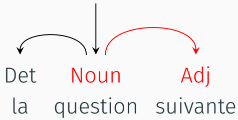
Det	Adj	Noun
a	big	apple

Det	Noun	Adj
la	question	suivante

Impact of word order

At data level:



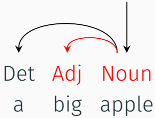


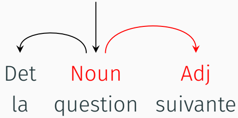
At model level:

$(s_0 = \textsc{Adj} \wedge n_0 = \textsc{Noun}) \Rightarrow \textsc{Left}$

$(s_0 = \textsc{Noun} \wedge n_0 = \textsc{Adj}) \Rightarrow \textsc{Right}$

At data level:



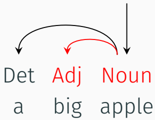


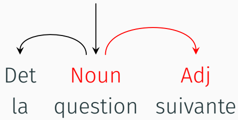
At model level: different features + different decisions

$(s_0 = \textsc{Adj} \wedge n_0 = \textsc{Noun}) \Rightarrow \textsc{Left}$ $(s_0 = \textsc{Noun} \wedge n_0 = \textsc{Adj}) \Rightarrow \textsc{Right}$

Impact of word order

At data level:





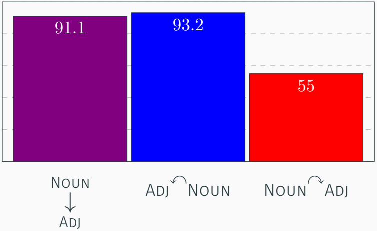
At model level:

different features + different decisions

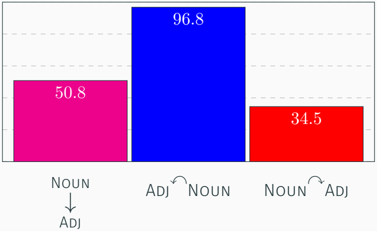
$(s_0 = \textsc{Adj} \land n_0 = \textsc{Noun}) \Rightarrow \textsc{Left}$ $\qquad$ $(s_0 = \textsc{Noun} \land n_0 = \textsc{Adj}) \Rightarrow \textsc{Right}$

On accuracy:

English → English



English → French



Impact of word order

At data level:

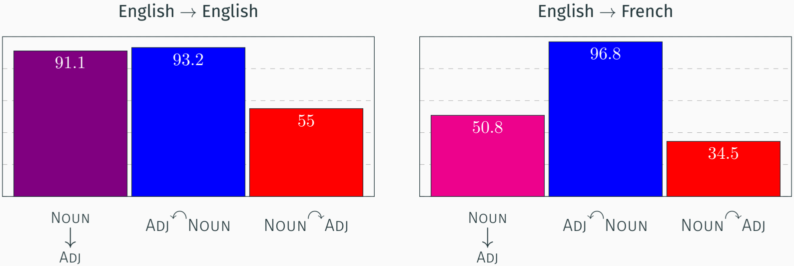
Det	Adj	Noun
a	big	apple

Det	Noun	Adj
la	question	suivante

At model level: different features + different decisions

$(s_0 = \textsc{Adj} \wedge n_0 = \textsc{Noun}) \Rightarrow \textsc{Left}$ $\qquad$ $(s_0 = \textsc{Noun} \wedge n_0 = \textsc{Adj}) \Rightarrow \textsc{Right}$

On accuracy: no knowledge sharing + transfer errors

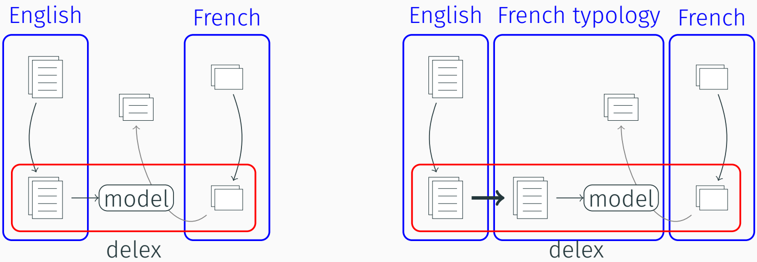


Leveraging WALS

Reshaping training instances

Our proposal:

preprocess source data, to match target regularities



↪ same dependency annotations

↪ same training process

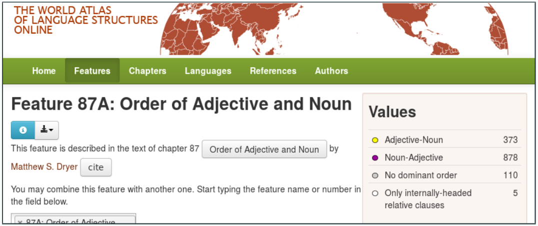
Reshaping training instances... a few examples

English – training data			French – desired output
Baseline		Our proposal	
Det Adj Noun a big apple	⇒	Det Noun Adj a apple big	Det Noun Adj la question suivante
Det Adj Noun the whole world	⇒	Det Noun Adj the world whole	Det Noun Adj la flotte romaine
Det Noun Noun an investment firm	⇒	Det Noun Noun an firm investment	Det Noun Adp Noun la plate-forme de distribution

WALS: a database of typological features for 2,679 languages

[`http://wals.info`]

↪ Over 1,000 languages with word order features



English	Adjective-Noun			
French	Noun-Adjective	Harris 1988: 227		

Using WALS

Heuristic rule extraction for **switching** and **deleting** tokens

$$\text{87A} \quad \begin{cases} \text{[English] Adjective-Noun} \\ \text{[French] Noun-Adjective} \end{cases}$$

$\Longrightarrow$ [English$\rightarrow$French] switch (most?) Adj-Noun into Noun-Adj

Using WALS

Heuristic rule extraction for **switching** and **deleting** tokens

$$\text{87A} \quad \begin{cases} \text{[English] Adjective-Noun} \\ \text{[French] Noun-Adjective} \end{cases}$$

$\Longrightarrow$ [English$\rightarrow$French] switch (most?) ADJ-NOUN into NOUN-ADJ

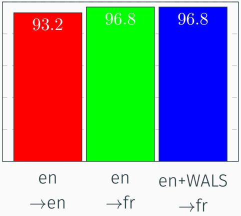
- ✓ easily extensible rule templates
  - $\hookrightarrow$ switch tokens, but also delete, insert, replace...
- ✓ accepts free order
- ✓ mostly conservative for related languages

- ✓ readily available for 1,000 languages
- ✓ most work already done by linguists

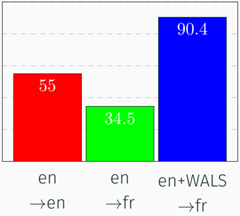
Experimental results on UD 1.3

English → French

ADJ ↶ NOUN

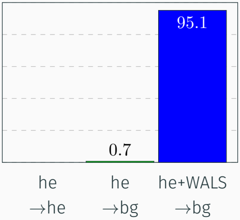


NOUN ↷ ADJ

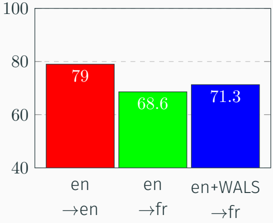


Hebrew → Bulgarian

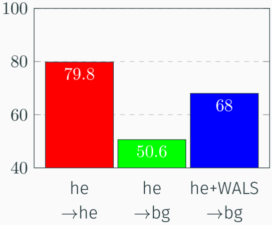
ADJ ↶ NOUN



Overall score: +2.7 UAS

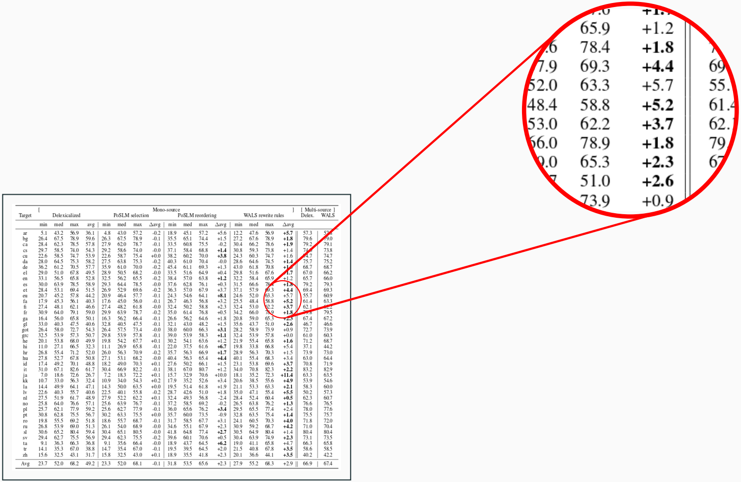


Overall score: +17.4 UAS



Experimental results on UD 1.3... over 40 languages

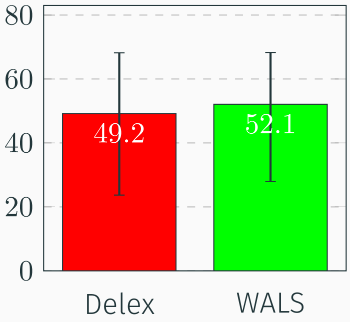
$\hookrightarrow$ 6,000+ experiments



Target	Delexicalized				PoSLM selection				PoSLM reordering				WALS rewrite rules				Multi-source Delex.	Multi-source WALS
	min	med	max	avg	min	med	max	Δavg	min	med	max	Δavg	min	med	max	Δavg		
ar	5.1	43.2	56.9	36.1	4.8	43.0	57.2	-0.2	18.9	45.1	57.2	+5.6	12.2	47.6	56.9	**+5.7**	57.3	[illegible]
bg	26.4	67.5	78.9	59.6	26.3	67.5	78.9	-0.1	35.5	65.1	74.4	+1.5	27.2	67.6	78.9	**+1.8**	79.6	79.0
ca	28.4	62.3	78.5	57.8	27.9	62.0	78.7	-0.1	33.5	60.8	75.5	-0.2	30.4	66.2	78.6	**+1.9**	79.2	79.1
cs	29.7	58.5	74.0	54.3	29.2	58.6	74.0	-0.0	37.1	58.4	68.8	**+1.4**	30.8	59.3	73.8	+1.4	[illegible]	73.8
cu	22.6	58.5	74.7	53.9	22.6	58.7	75.4	+0.0	38.2	60.2	70.0	**+3.8**	24.3	60.3	74.7	+1.6	[illegible]	74.7
da	28.0	64.5	75.3	58.2	27.5	63.8	75.3	-0.2	40.3	61.0	70.4	-0.0	28.6	64.6	74.5	**+1.4**	75.7	75.2
de	36.2	61.2	70.5	57.7	35.9	61.0	70.0	-0.2	45.4	61.1	69.3	+1.3	43.0	61.8	70.8	[illegible]	68.7	68.7
el	29.0	51.0	67.8	49.5	28.9	50.5	68.2	-0.0	33.5	51.6	64.9	+0.4	29.8	51.6	67.6	**+1.7**	67.0	66.2
en	33.1	56.5	65.8	52.8	32.5	56.2	65.5	-0.2	38.4	57.0	63.8	**+1.2**	32.2	58.4	65.9	+1.2	65.7	66.0
es	30.0	63.9	78.5	58.9	29.3	64.4	78.5	-0.0	37.6	62.8	76.1	+0.3	31.5	66.6	78.4	**+1.8**	79.2	79.3
et	28.4	53.1	69.4	51.5	26.9	52.9	69.6	-0.2	36.3	57.0	67.9	+3.7	37.1	57.9	69.3	**+4.4**	69.4	69.3
eu	20.7	45.2	57.8	44.2	20.9	46.4	57.7	-0.1	24.3	54.6	64.1	**+8.1**	24.6	52.0	63.3	+5.7	55.7	60.9
fa	17.9	45.3	56.1	40.3	17.6	45.0	56.0	-0.1	26.7	46.3	56.8	+3.2	25.5	48.4	58.8	**+5.2**	61.4	63.3
fi	27.4	48.1	62.1	46.6	27.4	48.2	61.8	-0.0	32.4	50.2	58.8	+2.3	32.4	53.0	62.2	**+3.7**	62.1	[illegible]
fr	30.9	64.0	79.1	59.0	29.9	63.9	78.7	-0.2	35.0	61.4	76.8	+0.5	34.2	66.0	78.9	**+1.8**	[illegible]	79.5
ga	16.4	56.0	65.8	50.1	16.3	56.2	66.4	-0.1	26.6	56.2	64.6	+1.8	20.8	59.0	65.3	**+2.3**	67.4	67.2
gl	33.0	40.3	47.5	40.6	32.8	40.5	47.5	-0.1	32.1	43.0	48.2	+1.5	35.6	43.7	51.0	**+2.6**	46.7	46.6
got	26.4	58.0	72.7	54.3	26.4	57.5	73.4	-0.0	38.0	60.0	66.3	**+3.1**	28.2	58.9	73.9	+0.9	72.7	73.9
grc	32.5	53.9	57.3	50.7	29.8	53.9	57.8	-0.1	39.0	53.9	58.3	**+1.1**	32.4	53.9	57.8	+0.0	61.0	60.3
he	20.1	53.8	68.0	49.9	19.8	54.2	67.7	+0.1	30.2	54.1	63.6	+1.2	21.9	55.4	65.8	**+1.6**	71.2	68.7
hi	11.0	27.1	66.5	32.3	11.1	26.9	65.8	-0.1	22.0	37.5	61.6	**+6.7**	19.8	33.8	66.8	+5.4	37.1	44.2
hr	26.8	55.4	71.2	52.0	26.0	56.3	70.9	-0.2	35.7	56.3	66.9	**+1.7**	28.9	56.3	70.3	+1.5	73.9	73.0
hu	27.8	52.7	67.8	50.8	27.1	53.1	68.2	-0.0	40.4	56.3	65.4	**+4.4**	40.1	55.4	68.3	+3.4	63.0	64.4
id	17.4	49.2	70.1	48.8	18.2	49.0	70.3	+0.1	27.6	50.2	66.1	+1.5	23.1	53.8	69.6	**+3.7**	70.8	71.9
it	31.0	67.1	82.6	61.7	30.4	66.9	82.2	-0.1	38.1	67.0	80.7	+1.2	34.0	70.8	82.3	**+2.2**	83.2	82.9
ja	7.0	18.6	72.6	26.7	7.2	18.3	72.2	+0.1	15.7	32.9	70.6	+10.0	18.1	35.2	72.3	**+11.4**	63.3	63.5
kk	10.7	33.0	56.3	32.4	10.9	34.0	54.3	+0.2	17.9	35.2	52.6	+3.4	20.6	38.5	55.6	**+4.9**	53.9	54.6
la	14.4	49.9	64.1	47.1	14.3	50.0	63.5	+0.0	19.5	51.4	61.8	+1.9	21.1	53.3	63.3	**+2.1**	58.3	60.0
lv	22.6	40.3	55.7	40.6	22.5	40.1	55.8	-0.2	28.7	42.6	51.0	+1.8	35.0	47.1	55.4	**+5.5**	50.2	57.3
nl	27.5	51.9	61.7	48.9	27.9	52.2	62.2	+0.1	32.4	49.3	56.8	-2.4	28.4	52.4	60.4	**+0.5**	62.3	60.7
no	25.8	64.0	76.6	57.1	25.6	63.9	76.7	-0.1	37.2	58.5	69.2	-0.2	26.5	63.8	76.2	**+1.3**	76.6	76.5
pl	25.7	62.1	77.9	59.2	25.6	62.7	77.9	-0.1	36.0	65.6	76.2	**+3.4**	29.5	65.5	77.4	+2.4	78.0	77.6
pt	30.8	62.8	75.5	56.7	30.2	63.3	75.5	+0.0	35.7	60.0	73.5	-0.9	32.8	63.5	75.4	**+1.4**	75.5	75.7
ro	19.8	55.5	69.2	51.8	18.6	55.7	68.7	-0.1	31.7	58.5	67.7	+3.1	24.1	60.5	70.3	**+4.0**	71.8	72.0
ru	26.8	53.9	69.0	51.3	26.1	54.0	68.9	-0.0	34.6	55.1	67.9	+2.3	30.9	59.2	68.7	**+4.2**	71.0	70.4
sl	30.6	65.2	80.4	59.4	30.4	65.1	80.5	-0.0	41.8	64.8	77.4	**+2.7**	30.5	64.9	80.4	+1.4	80.4	80.4
sv	29.4	62.7	75.5	56.9	29.4	62.3	75.5	-0.2	39.6	60.1	70.6	+0.5	30.4	63.9	74.9	**+2.3**	73.1	73.5
ta	9.1	36.3	66.3	36.8	9.1	35.6	66.4	-0.0	18.9	43.7	64.5	**+6.2**	19.0	41.1	65.8	+4.7	66.3	65.8
tr	14.1	35.3	67.0	38.8	14.7	35.4	67.0	-0.1	19.5	39.5	64.5	+2.0	21.5	40.8	67.8	**+3.5**	58.6	58.5
zh	15.6	32.5	43.1	31.7	15.8	32.5	43.0	+0.1	18.9	35.5	41.8	+2.3	20.1	36.6	44.1	**+3.5**	40.2	42.2
Avg	23.7	52.0	68.2	49.2	23.3	52.0	68.1	-0.1	31.8	53.5	65.6	+2.3	27.9	55.2	68.3	+2.9	66.9	67.4

Experimental results on UD 1.3... over 40 languages

↪ 6,000+ experiments



Average improvement:

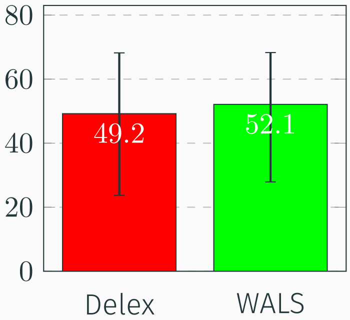
+2.9 UAS

+0.1 on best sources

+4.2 on worst sources

Experimental results on UD 1.3... over 40 languages

↪ 6,000+ experiments



Average improvement:

+2.9 UAS

+0.1 on best sources

+4.2 on worst sources

✓ Very efficient on nearly deterministic reorderings

✓ Specific error correction: many pairs (21%) have a very large (50+) improvement on at least one frequent tag pair

✓ Rarely detrimental

Leveraging raw data

Extract target regularities from raw data

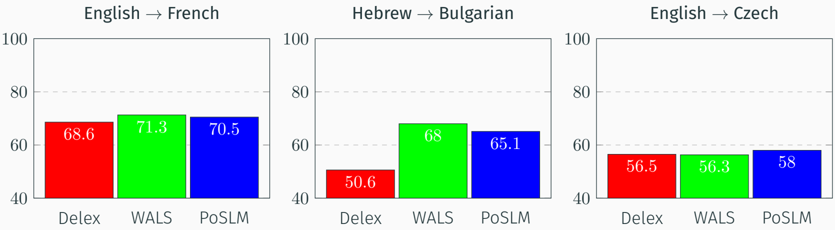
Contrastive experiment:
knowledge-driven method $\Longleftrightarrow$ data-driven method

$\hookrightarrow$ use a target language model to reorder the PoS sequence

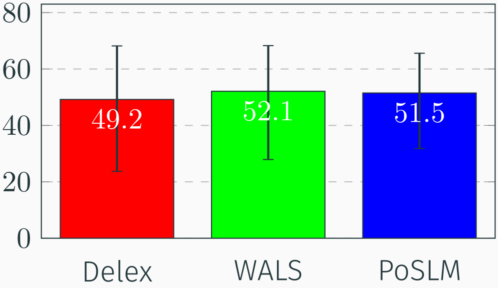
Procedure:

- train a PoSLM on delexicalized raw target data
- generate **local reordering lattices** for source data
- score them with the PoSLM
- keep the best projective reordering

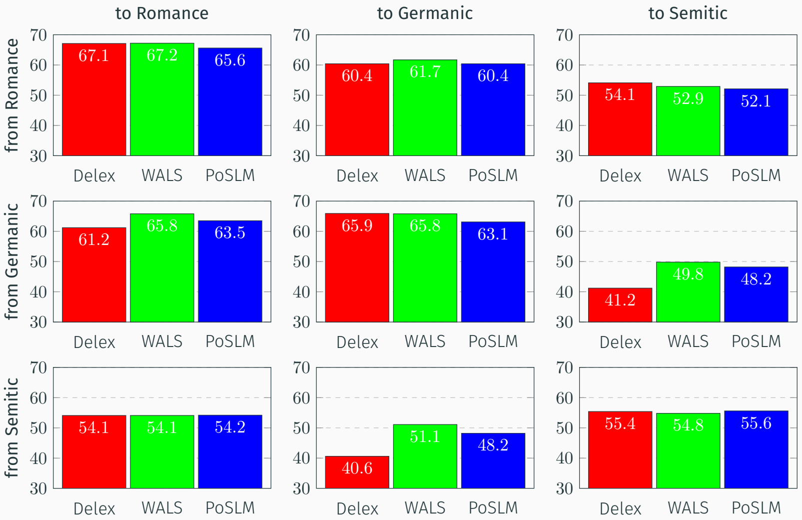
Comparative results



Average improvement: +2.3 UAS



Language family analysis



PoSLM approach: pros and cons

- ✗ can only reorder tokens
- ✗ does not acknowledge free order
- ✓ captures more diverse reordering patterns

- ✗ not *really* zero-resource
- ✓ *in theory* applicable to any language
- ✗ much more involved method

- ✗ too hard constraints for closely related sources
- ✗ sometimes out of control
- ✓ combine with known typology?

Wrap-up

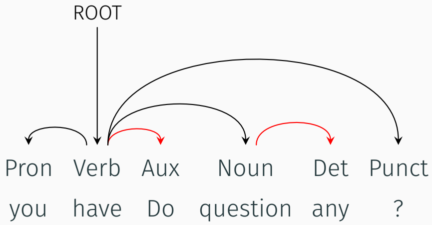
- **Zero-resource** scenario: target 1,000 languages, delexicalized cross-lingual transfer
- Despite common guidelines, **regular divergences** still affect representations, at model level
  $\Rightarrow$ poor knowledge sharing, transfer errors
- We reshape training examples, using **target word order**
  $\Rightarrow$ large improvements on specific errors
- Two setups: the naive **knowledge-rich approach** outperforms the involved **data-driven method**

- **Zero-resource** scenario: target 1,000 languages, delexicalized cross-lingual transfer

- Despite common guidelines, **regular divergences** still affect representations, at model level
  ⇒ poor knowledge sharing, transfer errors

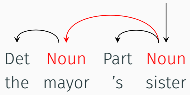
- We reshape training examples, using **target word order**
  ⇒ large improvements on specific errors

- Two setups: the naive **knowledge-rich approach** outperforms the involved **data-driven method**


ROOT
Pron Verb Aux Noun Det Punct
you have Do question any ?

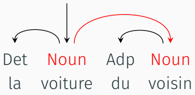
Appendix: more examples

English



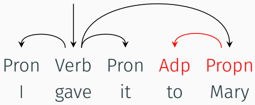
$$(s_0 = \textsc{Noun} \wedge n_0 = \textsc{Noun}) \Rightarrow \textsc{Left}$$

$\longrightarrow$ French

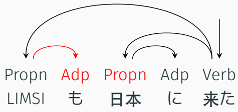


$$(s_0 = \textsc{Noun} \wedge n_0 = \textsc{Noun}) \Rightarrow \textsc{Right}$$

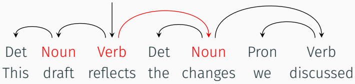
English



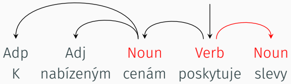
⟶ Japanese



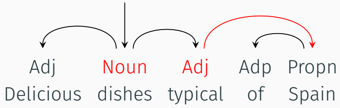
English

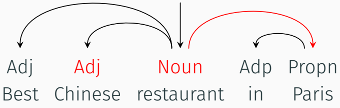


⟶ Czech



Limits of feature engineering





Full abstraction from word order is not desirable