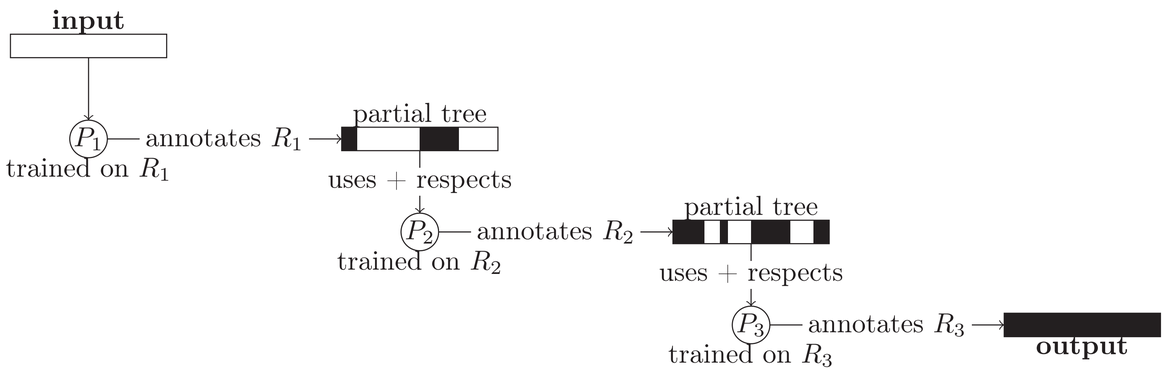
LIMSI@CoNLL'17: UD Shared Task

Lauriane Aufrant[1,2], *Guillaume Wisniewski*[1]

(1) LIMSI, CNRS, Univ. Paris-Sud, Université Paris-Saclay, (2) DGA

{lauriane.aufrant, guillaume.wisniewski}@limsi.fr

A new combination method: cascade parsers



P_1 = annotation projection — R_1 = Adj-Noun, Det-Noun...

P_2 = delexicalized transfer — R_2 = Noun-Verb, Adp-Noun...

P_3 = monolingual parser — R_3 = all remaining dependencies

Procedure:

- Train several base parsers (eg. P_1, P_2, P_3) on various resources
- Evaluate their **competence regions** (eg. R_1, R_2, R_3): attribute each dependency type to the parser that annotates it best
- **Retrain** each parser in turn: only on its region, and **using the outputs of the previous parsers** as enriched input
- At test time: annotate the input with each parser in turn, producing **partial trees** that grow at each step

↪ **A small resource is better leveraged on an already enriched input.**

Strategy for the shared task

Using development data, we compare and submit different parsers for each treebank. Depending on the available resources, the best parser may differ.

Competition between:

- Monolingual parsers: UDPipe (neural), PanParser (perceptron), Delex
- Cross-lingual parsers:
  - X-Delex, using delexicalization and WALS rewrite rules
  - Project, using annotation projection of partial trees
  - Multi-source Delex, using KL_{cpos^3} language similarity
- Cascade parsers: UDPipe+PanParser, Project+X-Delex+PanParser...
  ↪ **Diversity compensates for small/incomplete data**

Source languages for cross-lingual parsers are chosen heuristically, based on WALS features, KL_{cpos^3} language similarity and treebank sizes.

Relation labels are predicted in a second step, with similar combination methods.

WALS rewrite rules: see [Aufrant et al., 2016]
Partial projection: see [Lacroix et al., 2016]
KL_{cpos^3} language similarity: see [Rosa and Zabokrtsky, 2015]

Overall results

	UDPipe [off.]		LIMSI [off.]		LIMSI [unoff.]	
	F1/LAS	Rank	F1/LAS	Rank	F1/LAS	Rank
Tokenization	98.77	8	**98.95**	1	98.95	
All tags	73.74	4	**73.86**	2	73.86	
All treebanks	68.35	13	67.72	17	**68.90**	12
Big (55)	73.04	17	**73.64**	13	73.64	
PUD (14)	68.33	13	62.24	26	**69.07**	
Small (8)	**51.80**	15	51.71	16	51.71	
Surprise (4)	37.07	11	**37.57**	9	37.57	

Main conclusions:

- **Cascading proves successful in low-resource settings** but still faces reliability challenges: lack of confidence mechanisms, unreliability of delexicalized models when PoS accuracies are low...
- **Tuning/development data** remain a bottleneck of our method, both for accurate estimation of competence regions and for model selection.
- End-to-end parsing rewards a lot **good tokenization**.
- Our ranking is mostly penalized by **huge unexpected drops** on the PUD treebanks (cf discussion).

Per-treebank results: comparison with the UDPipe baseline

Languages with custom tokenization:

- Japanese: +4.14% on tokenization (using KyTea) ⇒ +7.80 LAS
- Chinese: +2.44% on tokenization (using KyTea) ⇒ +2.58 LAS
- Vietnamese: +4.83% on tokenization (by postprocessing multi-token words) ⇒ +4.55 LAS

PUD treebanks: submission of one of the models trained on this language

- Official run (using the treebank with best LAS on own devset): -6.09 LAS
- Unofficial run (using always the main treebank, as the baseline did): +0.74 LAS

Other treebanks (including surprise ones):

Treebank size (#sentences)	# languages > baseline	# languages < baseline	# baseline submissions	Avg gain (LAS)
> 10,000	3	2	15	+0.06
1,000-10,000	18	5	9	+0.44
< 1,000	10	2	0	+0.33

⟹ Our strategy is most effective on small treebanks.

KyTea: see [Neubig et al., 2011]

Discussion: PUD and cross-treebank consistency

The shared task results unveiled huge drops on the PUD treebanks, depending on the training treebank, for the same model and language.
↪ **They are not entirely explained by treebank size or domain.**

Experiments with UDPipe: preprocess with the model from treebank A, parse with the model from treebank B, evaluate on the PUD treebank.

Tok/tag	Parser	LAS on PUD
en	en	78.95
en_lines	en	63.42
en	en_lines	47.30
en_lines	en_lines	64.28
ru	ru	68.31
ru_syn.	ru	68.18
ru	ru_syn.	52.36
ru_syn.	ru_syn.	59.87

Tok/tag	Parser	LAS on PUD
fi	fi	78.65
fi_ftb	fi	52.80
fi	fi_ftb	44.99
fi_ftb	fi_ftb	47.27
sv	sv	70.62
sv_lines	sv	51.63
sv	sv_lines	49.41
sv_lines	sv_lines	65.11

Detection of some annotation issues:

- Multi-token words preannotated with '_' (`el`, `fi_ftb`, `ru_syntagrus`)
- Raw text is already tokenized (`da`, `fi_ftb`)
- Incompatible preprocessing among treebanks (English, Portuguese, Swedish)
- Possibly, incompatible parsing schemes (Finnish, Russian, Spanish...)