RobertNLP at the 2021 IWPT Shared Task: Simple Enhanced UD Parsing for 17 Languages

Stefan Grünewald[1,2], Frederik Oertel[2], Annemarie Friedrich[2]

[1]Institut für Maschinelle Sprachverarbeitung, University of Stuttgart
[2]Bosch Center for Artificial Intelligence, Renningen, Germany

IWPT21@ACL



Institut für Maschinelle Sprachverarbeitung

BOSCH

Overview
End-to-End Enhanced Graph Parsing for English

- **Task:** Parsing from raw text into Enhanced UD
  - Multilingual: 17 languages, 29 corpora
- **Our submission:**
  - Adapted from our English-only submission for IWPT20
  - **3rd** place overall

Team	ELAS Score
TGIF	89.24
ShanghaiTech	87.07
RobertNLP	**86.97**
Combo	83.79
Unipi	83.64
DCU-EPFL	83.57
Grew	81.58
FastParse	65.81
NUIG	30.03

Official results (ELAS, IWPT21 test set)

Overview
End-to-End Enhanced Graph Parsing for English

- **Task:** Parsing from raw text into Enhanced UD
  - Multilingual: 17 languages, 29 corpora
- **Our submission:**
  - Adapted from our English-only submission for IWPT20
  - **3rd** place overall

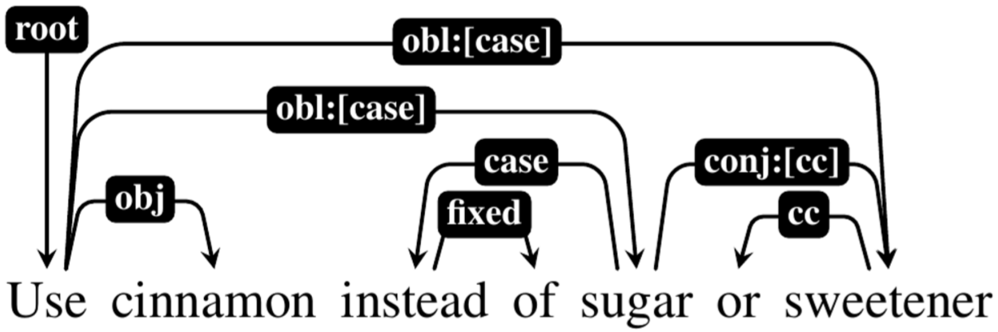
- **Our approach:**
  1. Classify dependencies between pairs of tokens
  2. Ensemble predictions of several models and build valid dependency graphs
  3. Lexicalize dependency labels via a hybrid approach

Team	ELAS Score
TGIF	89.24
ShanghaiTech	87.07
RobertNLP	**86.97**
Combo	83.79
Unipi	83.64
DCU-EPFL	83.57
Grew	81.58
FastParse	65.81
NUIG	30.03

Official results (ELAS, IWPT21 test set)

System Overview
Dependency Classification

- Predict the dependency relation for every pair of tokens [Dozat & Manning, 2018]
  - Treat non-existence of a dependency as simply another label (∅)
  - **Unfactorized** system



	[root]	Use	cinnamon	instead	of	sugar	or	sweetener
[root]	∅	*root*	∅	∅	∅	∅	∅	∅
Use	∅	∅	*obj*	∅	∅	*obl:[case]*	∅	*obl:[case]*
cinnamon	∅	∅	∅	∅	∅	∅	∅	∅
instead	∅	∅	∅	∅	*fixed*	∅	∅	∅
of	∅	∅	∅	∅	∅	∅	∅	∅
sugar	∅	∅	∅	*case*	∅	∅	∅	*conj:[cc]*
or	∅	∅	∅	∅	∅	∅	∅	∅
sweetener	∅	∅	∅	∅	∅	∅	*cc*	∅

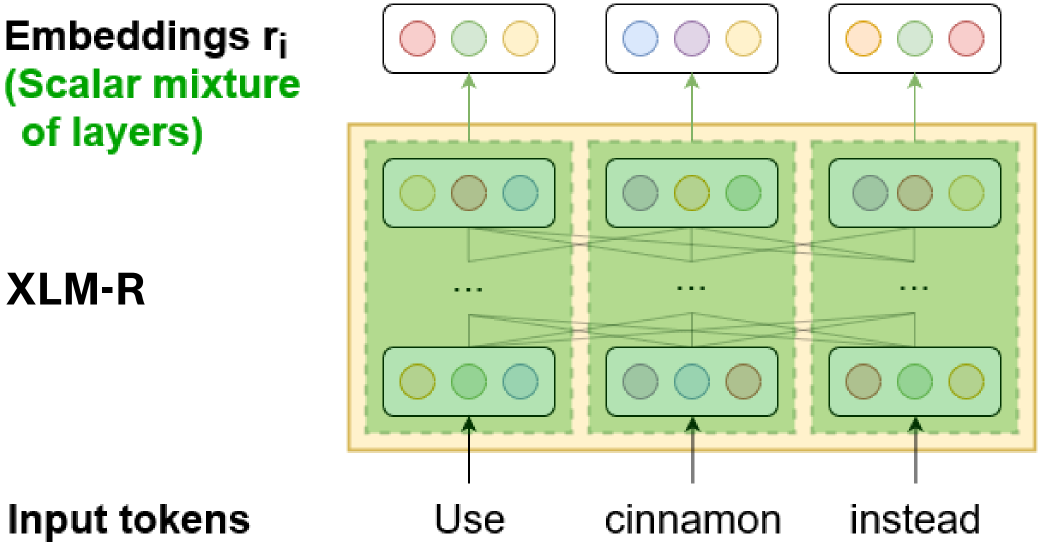
System Overview
Dependency Classification

System Overview
Dependency Classification

Input tokens Use cinnamon instead

Trankit-large for tokenization and sentence segmentation [Nguyen et al., 2021]

System Overview
Dependency Classification

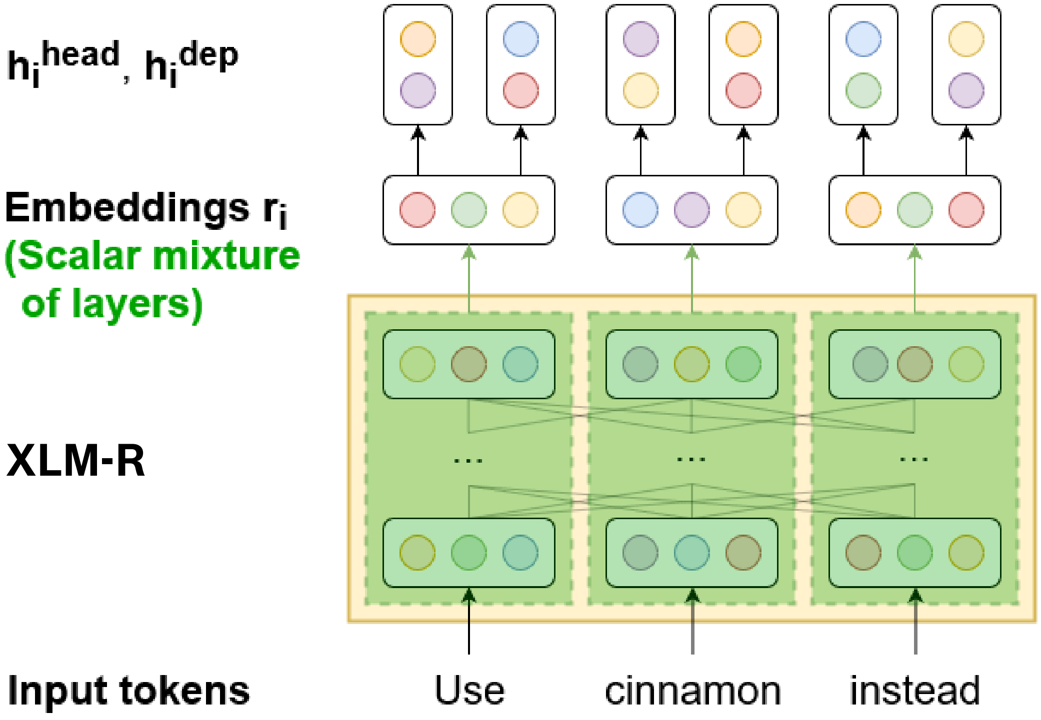


Contextualized embeddings from a weighted sum of **XLM-R** layers [Conneau et al., 2020]

- [root] → learned embedding

Trankit-large for tokenization and sentence segmentation [Nguyen et al., 2021]

System Overview
Dependency Classification



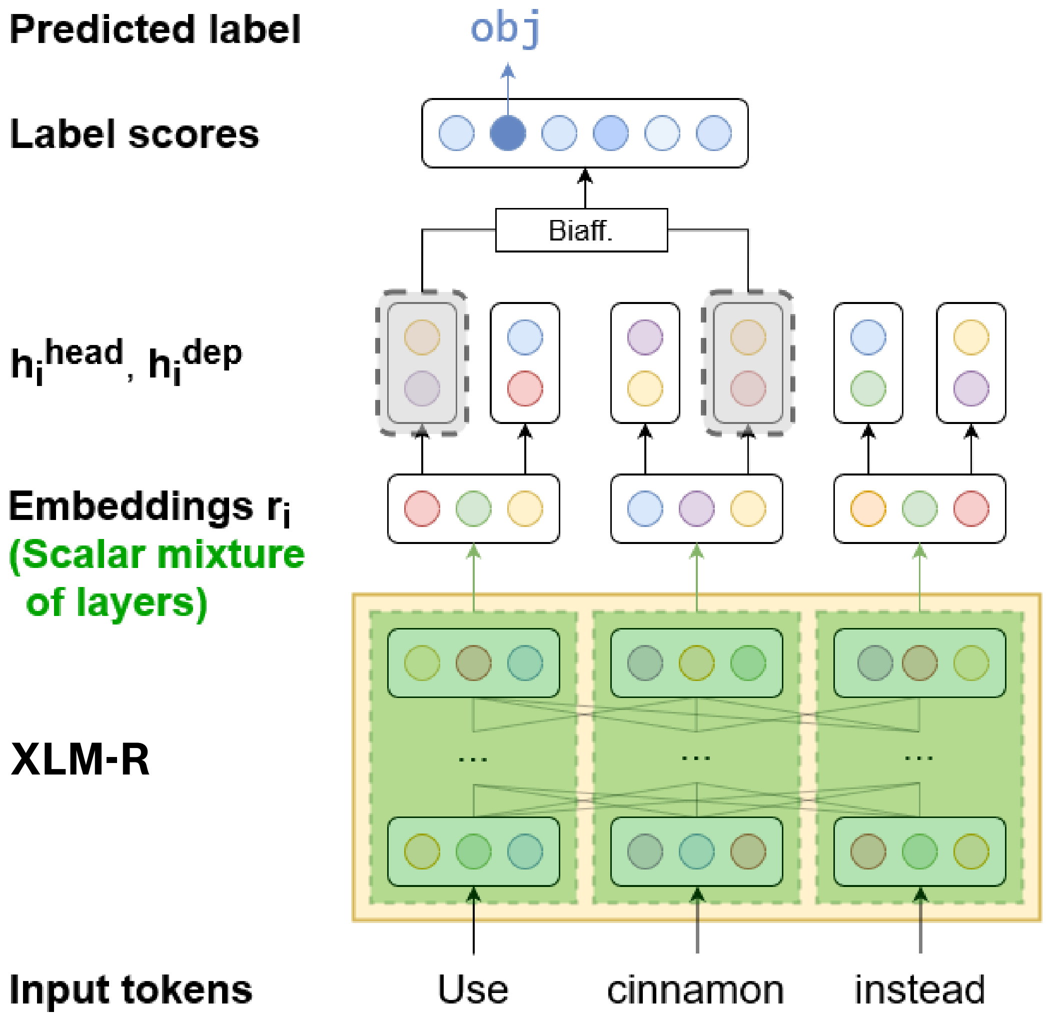
Each token receives a **head** and a **dependent** representation

Contextualized embeddings from a weighted sum of **XLM-R** layers [Conneau et al., 2020]

- [root] → learned embedding

Trankit-large for tokenization and sentence segmentation [Nguyen et al., 2021]

System Overview
Dependency Classification



Biaffine classifier: Probabilities for the different dependency labels for each head/dependent pair in the sentence [Dozat & Manning, 2017]

Each token receives a **head** and a **dependent** representation

Contextualized embeddings from a weighted sum of **XLM-R** layers [Conneau et al., 2020]

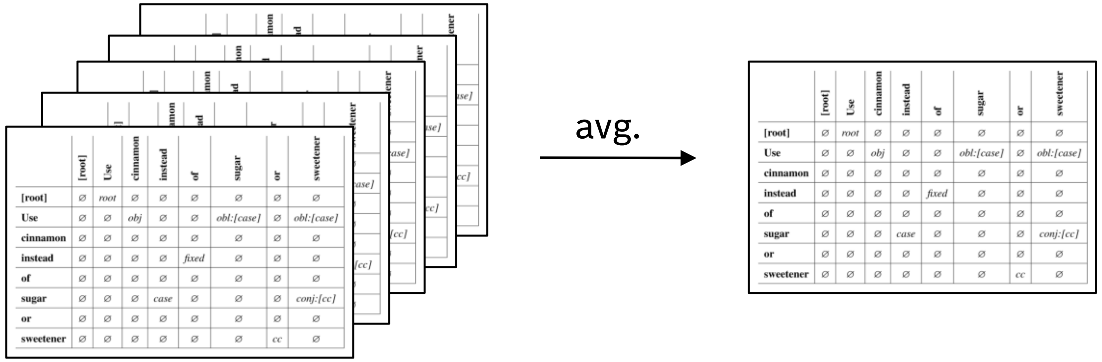
- [root] → learned embedding

Trankit-large for tokenization and sentence segmentation [Nguyen et al., 2021]

Institut für Maschinelle Sprachverarbeitung
BOSCH

System Overview
Ensembling Predictions

- To further improve accuracy, we train 5 models per language and average their predictions (class probabilities)
- For languages with multiple treebanks (e.g. Czech), we mix models trained on different treebanks to increase robustness
- However, accuracy improves even when only ensembling models trained on the same data



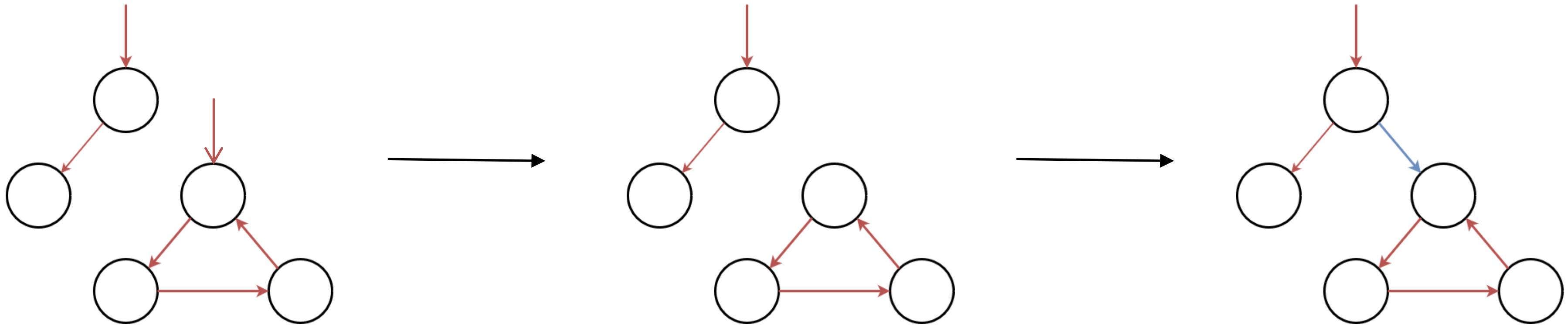
Language	Best single	Ensemble
Arabic	81.37	81.58
Czech	89.99	90.21
English	87.29	87.88
Finnish	90.77	91.01
Tamil	58.24	59.33

ELAS F1 (IWPT21 test)

System Overview
Assembling the Dependency Graph

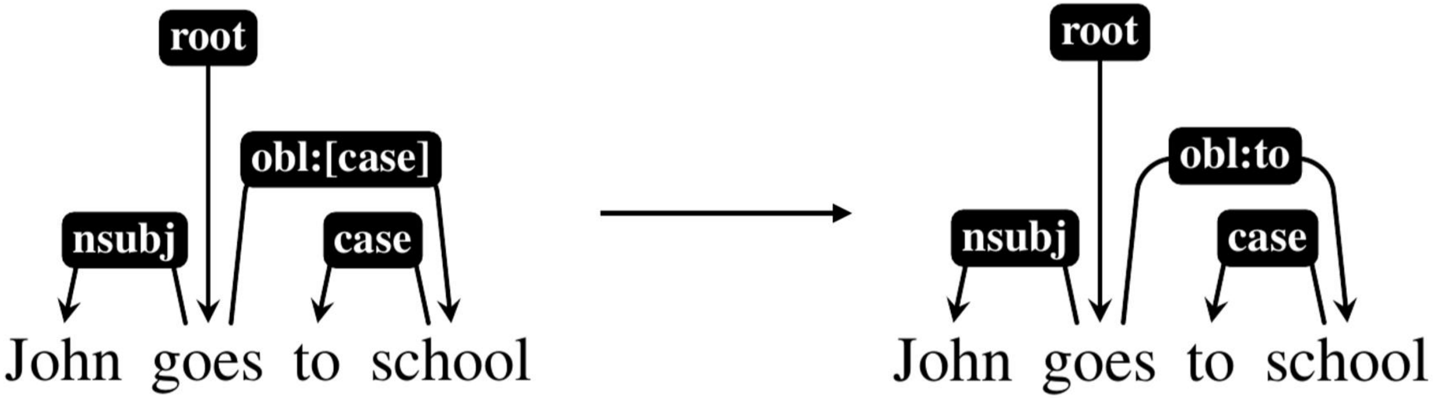
- The **union of all predicted edges** forms a dependency graph
- But: For enhanced UD graphs to be valid, all tokens must be reachable from the root
  - Because our classifier considers each edge in isolation, it cannot guarantee this
- To ensure graph validity, we perform two heuristic steps:

1. If there are multiple tokens designated as root, remove all but the most confidently predicted one
2. As long as there are nodes in the graph that are not reachable from the root, greedily add the most confidently predicted non-∅ dependency from a reachable to an unreachable node



System Overview
Label lexicalization: Heuristic

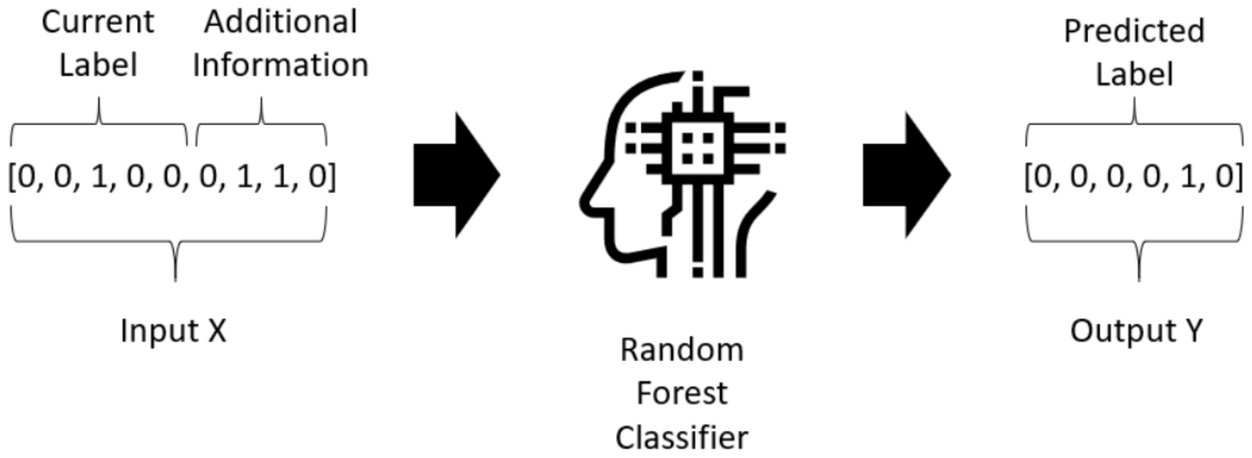
- Problem: Lexicalized labels (e.g. *obl:to*) → data sparsity
- Solution: Train/predict placeholder labels first, then re-lexicalize
  - Placeholder labels: *obl:[case]*, *nmod:[case]*, *acl:[mark]*, *advcl:[mark]*, *conj:[cc]*
- Last year's strategy (optimized for English): Rule-based heuristic for re-lexicalization



Institut für Maschinelle Sprachverarbeitung

System Overview
Label lexicalization: ML Transducer

- Heuristic works well for English and some other languages, but much worse for the rest
  - Reasons: Lack of lemmatization, different handling of multiword expressions
- Solution: Write a machine learning system that detects and corrects incorrect lexicalizations based on sentence context after the heuristic was run



{obl:te, om, zonder} ⟶ {obl:om}

System Overview

Label lexicalization: ML Transducer in Hybrid Setup

- Hybrid Setup:

Heuristic + ML Transducer

Heuristic	Transducer	→	Hybrid
nmod:from	nmod:from	→	nmod:from
obl:te	obl:om	→	obl:om
conj:in	acl:in	→	conj:in

Treebank	Heuristic	Hybrid
Arabic-PADT	93.4	97.5
Czech-PDT	90.9	99.2
English-EWT	98.4	98.8
Estonian-EDT	98.8	99.8
Latvian-LVTB	99.4	99.7
Polish-PDB	91.8	98.9
Slovak-SNK	93.0	98.0
Tamil-TTB	16.1	66.1

Re-leixcalization accuracy (%)

Experiments
Setup

- Model implemented using **PyTorch,** [Paske et al., 2019] **HuggingFace Transformers,** [Wolf et al., 2019] and **Scikit-learn** [Pedregosa et al., 2011]
- Training and validation on IWPT data
- Training takes 1—24 hours (depending on treebank) on a single nVidia Tesla V100 GPU
- Hyperparameters: → Paper

Institut für Maschinelle Sprachverarbeitung
BOSCH

Experiments
Results

Official IWPT 2021 result:
Avg. ELAS F1 = 86.97%

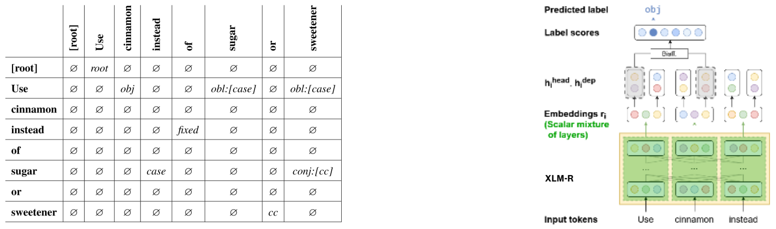
- High parsing accuracy overall, outperforming the median on all languages
- Ensembling and ML-based re-lexicalization help, but system would still achieve 3rd place without them
- Best language: Italian (93.28)
- Worst language: Tamil (59.33)

Language	TGIF	Median	RobertNLP
Average	**89.24**	**83.64**	**86.97**

Language	TGIF	Median	RobertNLP
Arabic	81.23	76.39	81.58
Bulgarian	93.63	90.84	93.16
Czech	92.24	89.08	90.21
Dutch	91.78	84.14	88.37
English	88.19	85.70	87.88
Estonian	88.38	84.02	86.55
Finnish	91.75	89.02	91.01
French	91.63	87.32	88.51
Italian	93.31	91.81	93.28
Latvian	90.23	84.57	88.82
Lithuanian	86.06	78.04	80.76
Polish	91.46	88.31	89.78
Russian	94.01	90.90	92.64
Slovak	94.96	87.04	89.66
Swedish	89.90	84.91	88.03
Tamil	65.58	52.27	59.33
Ukrainian	92.78	86.92	88.86
Average	**89.24**	**83.64**	**86.97**

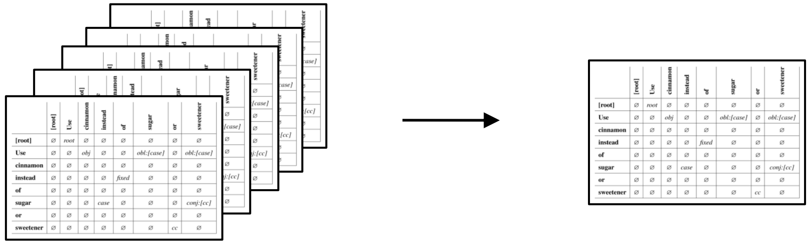
Conclusion & Future Work

- **RobertNLP:** A simple yet effective method to parse raw text into Enhanced Universal Dependencies

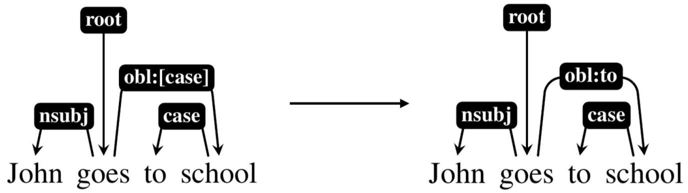


	[root]	Use	cinnamon	instead	of	sugar	or	sweetener
[root]	∅	*root*	∅	∅	∅	∅	∅	∅
Use	∅	∅	*obj*	∅	∅	*obl:[case]*	∅	*obl:[case]*
cinnamon	∅	∅	∅	∅	∅	∅	∅	∅
instead	∅	∅	∅	∅	*fixed*	∅	∅	∅
of	∅	∅	∅	∅	∅	∅	∅	∅
sugar	∅	∅	∅	*case*	∅	∅	∅	*conj:[cc]*
or	∅	∅	∅	∅	∅	∅	∅	∅
sweetener	∅	∅	∅	∅	∅	∅	*cc*	∅

1. Predict the best relation for each pair of tokens



2. Ensemble predictions



3. Hybrid re-lexicalization strategy

Future work:

- Improve performance in low-resource scenarios such as Tamil

Institut für Maschinelle Sprachverarbeitung
BOSCH

References

- Timothy Dozat and Christopher D. Manning (2017): **Deep biaffine attention for neural dependency parsing.** In: 5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings.
- Timothy Dozat and Christopher D. Manning (2018): **Simpler but more accurate semantic dependency parsing.** In: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), pages 484– 490.
- Dan Kondratyuk and Milan Straka (2019): **75 languages, 1 model: Parsing universal dependencies universally.** In: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), pages 2779–2795.
- Adam Paszke et al. (2019): **PyTorch: An imperative style, high-performance deep learning library.** In: Advances in Neural Information Processing Systems 32, pages 8024–8035.
- Thomas Wolf et al. (2019): **Transformers: State-of-the-art Natural Language Processing.** arXiv preprint arXiv:1910.03771.

Institut für Maschinelle Sprachverarbeitung

BOSCH