RobertNLP at the IWPT 2020 Shared Task: Surprisingly Simple Enhanced UD Parsing for English

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

[1]Institut für Maschinelle Sprachverarbeitung, University of Stuttgart

[2]Bosch Center for Artificial Intelligence, Renningen, Germany

https://github.com/boschresearch/robertnlp-enhanced-ud-parser

IWPT 2020 @ ACL

Overview
End-to-End Enhanced Graph Parsing for English

- **Our submission:**
  - 1st place on English test data
  - As of now, English only

Submission	ELAS F1
RobertNLP	**88.94**
TurkuNLP	87.15
median	83.41
UDify + converter	85.67

Official results for English (IWPT test)

BOSCH

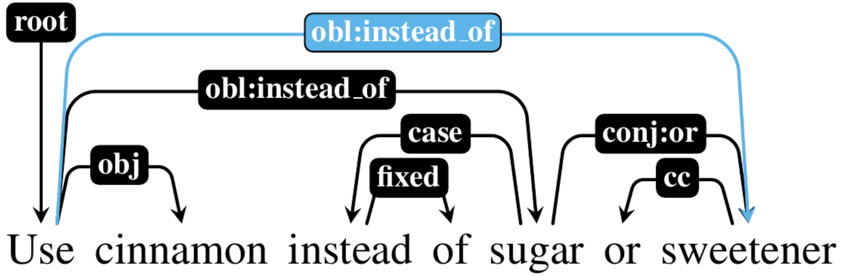
Overview
End-to-End Enhanced Graph Parsing for English

- **Our submission:**
  - 1st place on English test data
  - As of now, English only

Submission	ELAS F1
RobertNLP	**88.94**
TurkuNLP	87.15
median	83.41
UDify + converter	85.67

Official results for English (IWPT test)

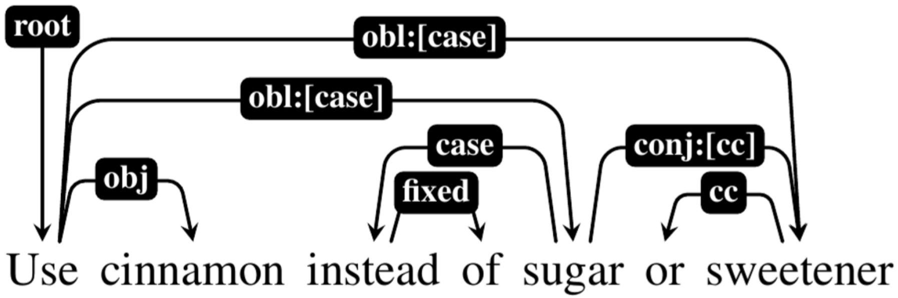
- **Our approach:**
  1. Predict the whole dependency graph directly
  2. Ensure graph validity
  3. Lexicalize labels



Grünewald & Friedrich, IWPT 2020 | 09.07.2020

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 (∅)



	[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*	∅

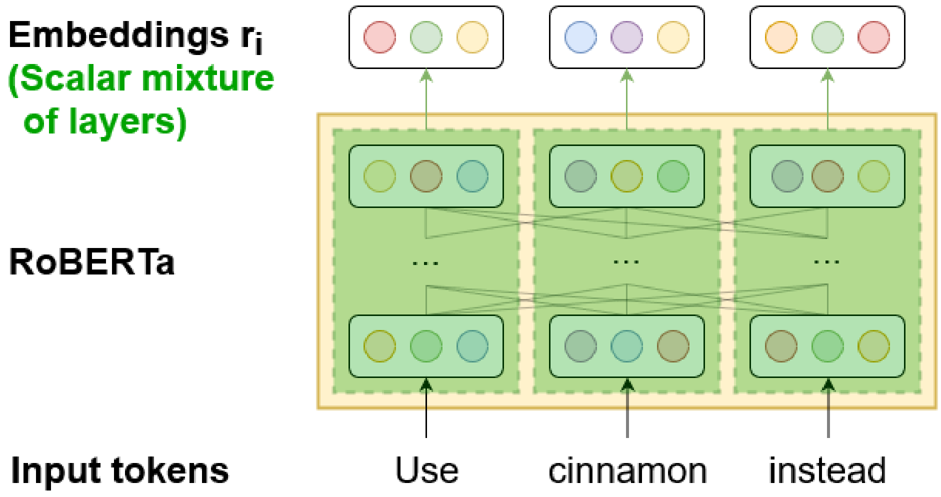
System Overview
Dependency Classification

System Overview
Dependency Classification

Input tokens Use cinnamon instead

StanfordNLP for tokenization and sentence segmentation

System Overview
Dependency Classification

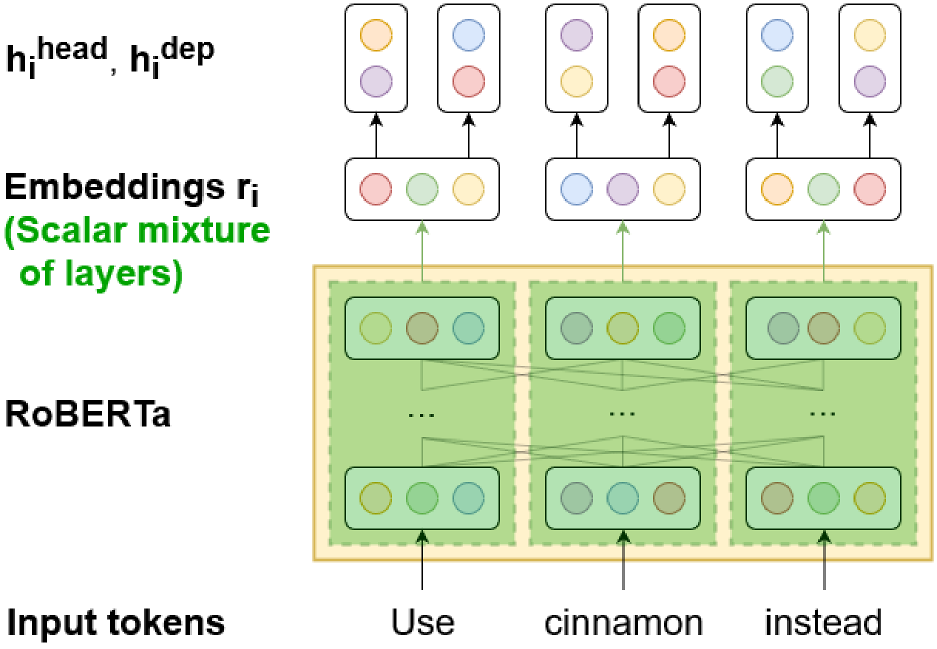


Contextualized embeddings from a weighted sum of **RoBERTa** layers [Kondratyuk & Straka, 2019]

- [root] → learned embedding

StanfordNLP for tokenization and sentence segmentation

System Overview
Dependency Classification



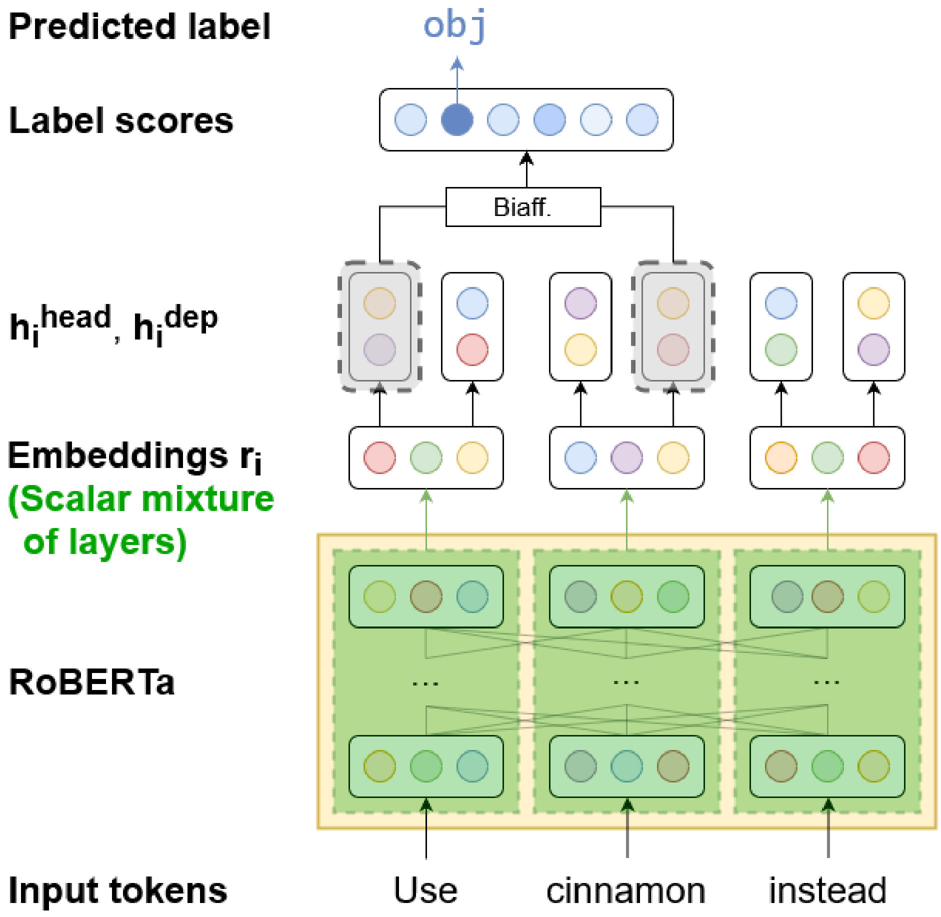
Each token receives a **head** and a **dependent** representation

Contextualized embeddings from a weighted sum of **RoBERTa** layers [Kondratyuk & Straka, 2019]

- [root] → learned embedding

StanfordNLP for tokenization and sentence segmentation

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 **RoBERTa** layers [Kondratyuk & Straka, 2019]

- [root] → learned embedding

StanfordNLP for tokenization and sentence segmentation

BOSCH

System Overview
Ensuring Graph Validity

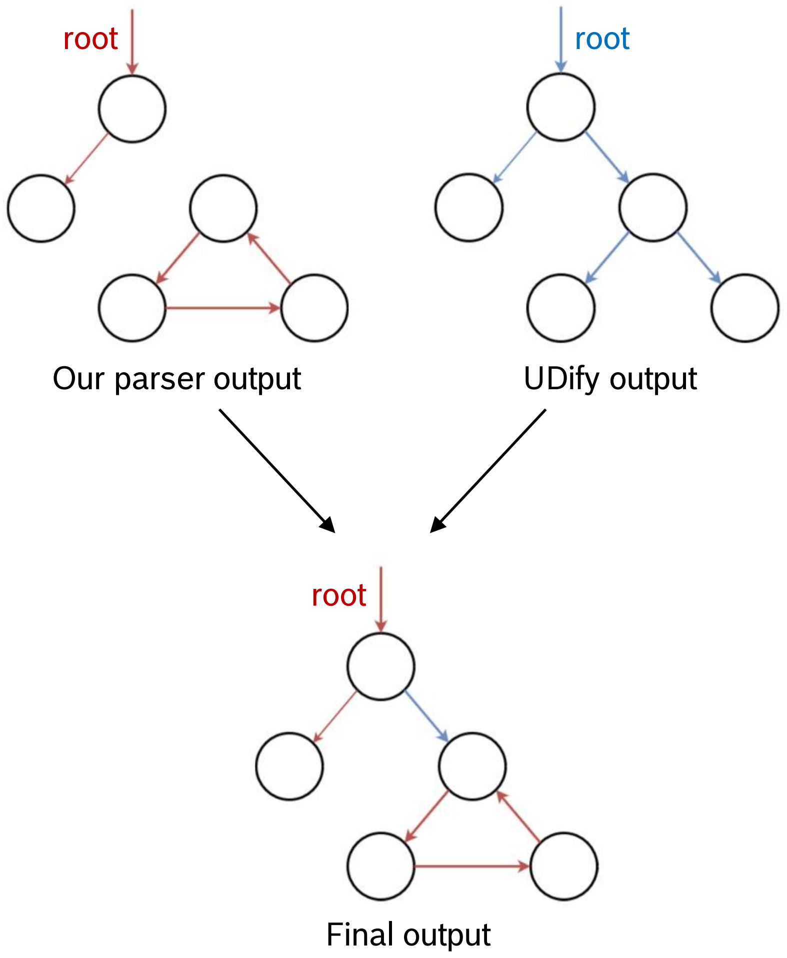
- The **union of all predicted edges** forms a dependency graph
- 99% of graphs are structurally valid
  - All nodes are reachable from the root

System Overview
Ensuring Graph Validity

- The **union of all predicted edges** forms a dependency graph
- 99% of graphs are structurally valid
  - All nodes are reachable from the root
- For the remaining 1% invalid graphs:
  1. For tokens lacking a head, assign to them the highest-scoring incoming non-∅ dependency

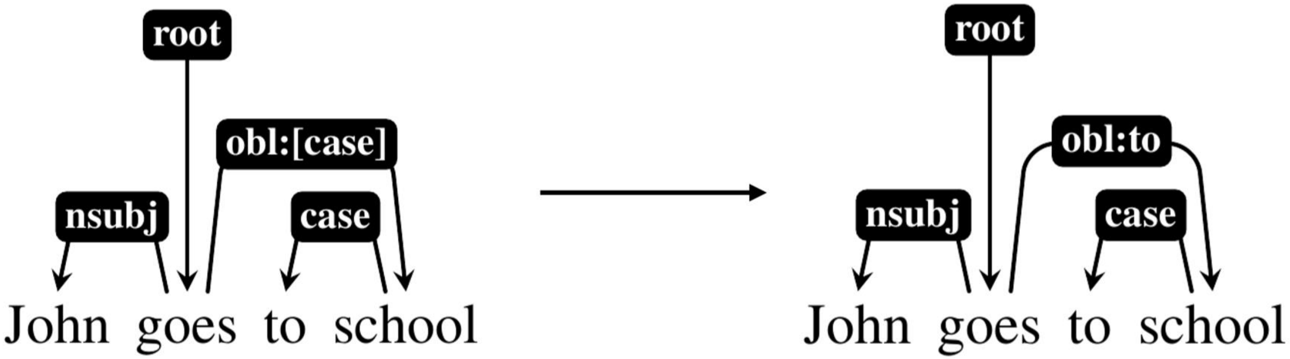
System Overview
Ensuring Graph Validity

- The **union of all predicted edges** forms a dependency graph
- 99% of graphs are structurally valid
  - All nodes are reachable from the root
- For the remaining 1% invalid graphs:
  1. For tokens lacking a head, assign to them the highest-scoring incoming non-∅ dependency
  2. If there are still unreachable nodes, heuristically add edges from a basic dependency tree (produced by the external UDify parser) until the graph is connected



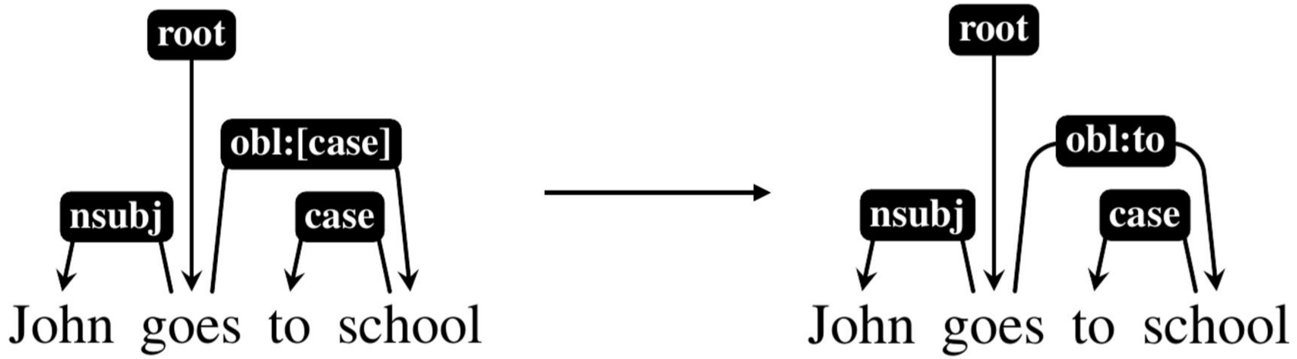
System Overview
Label lexicalization

- Problem: Lexicalized labels (e.g. *obl:to*) → data sparsity
- Solution: Predict placeholder labels first, then re-lexicalize based on a set of rules (e.g. *obl:[case]*)



System Overview
Label lexicalization

- Problem: Lexicalized labels (e.g. *obl:to*) → data sparsity
- Solution: Predict placeholder labels first, then re-lexicalize based on a set of rules (e.g. *obl:[case]*)



- Slightly more complex rules for multiword expressions, coordination, and enumerations → Paper
- If base relation was predicted correctly, we find the correct lexical material in 98.4% of cases

Experiments
Setup

- Model implemented using **PyTorch** [Paske et al., 2019] and **HuggingFace Transformers** [Wolf et al., 2019]
- Training and validation on the EWT corpus
- Around 9 hours of training on a single nVidia Tesla V100 GPU
- Hyperparameters: → Paper
- Code: `https://github.com/boschresearch/robertnlp-enhanced-ud-parser`

BOSCH

Analysis & Experimental Results

Official IWPT 2020 result (English):
ELAS F1 = 88.94%

- 90.80% F1 when using gold tokenization/segmentation
- Recall is considerably lower on relations exclusive to the enhanced layer as opposed to relations also present in the basic layer (83.64% vs. 91.60%)
- Certain label types (e.g. *punct, flat, compound*) remain problematic → Paper

Analysis & Experimental Results

Official IWPT 2020 result (English):
ELAS F1 = 88.94%

- 90.80% F1 when using gold tokenization/segmentation
- Recall is considerably lower on relations exclusive to the enhanced layer as opposed to relations also present in the basic layer (83.64% vs. 91.60%)
- Certain label types (e.g. *punct, flat, compound*) remain problematic → Paper

Additional research questions:

Analysis & Experimental Results

Official IWPT 2020 result (English):
ELAS F1 = 88.94%

- 90.80% F1 when using gold tokenization/segmentation
- Recall is considerably lower on relations exclusive to the enhanced layer as opposed to relations also present in the basic layer (83.64% vs. 91.60%)
- Certain label types (e.g. *punct, flat, compound*) remain problematic → Paper

Additional research questions:

- Which pre-trained LM works best?

Embeddings	Train	ELAS F1
BERT-base	EWT	87.49
RoBERTa-base	EWT	88.17
BERT-large	EWT	88.18
RoBERTa-large	**EWT**	**88.94**

Analysis & Experimental Results

Official IWPT 2020 result (English):
ELAS F1 = 88.94%

- 90.80% F1 when using gold tokenization/segmentation
- Recall is considerably lower on relations exclusive to the enhanced layer as opposed to relations also present in the basic layer (83.64% vs. 91.60%)
- Certain label types (e.g. *punct, flat, compound*) remain problematic → Paper

Additional research questions:

- Which pre-trained LM works best?
- Does incorporating additional English UD corpora as training data help?

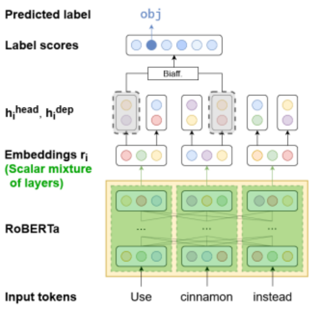
Embeddings	Train	ELAS F1
BERT-base	EWT	87.49
RoBERTa-base	EWT	88.17
BERT-large	EWT	88.18
RoBERTa-large	**EWT**	**88.94**
RoBERTa-large	UD2.5	87.85

Conclusion & Future Work

https://github.com/boschresearch/robertnlp-enhanced-ud-parser

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

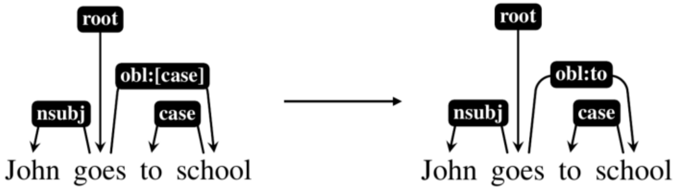
	[root]	Use	cinnamon	instead	of	sugar	or	sweetener
[root]	∅	*root*	∅	∅	∅	∅	∅	∅
Use	∅	∅	*obj*	∅	∅	*obl:[case]*	∅	*obl:[case]*
cinnamon	∅	∅	∅	∅	∅	∅	∅	∅
instead	∅	∅	∅	∅	*fixed*	∅	∅	∅
of	∅	∅	∅	∅	∅	∅	∅	∅
sugar	∅	∅	∅	*case*	∅	∅	∅	*conj:[cc]*
or	∅	∅	∅	∅	∅	∅	∅	∅
sweetener	∅	∅	∅	∅	∅	∅	*cc*	∅



Predict the best relation for each pair of tokens

Future work:

- Adapt our model to other languages
  - Requires some manual work due to label lexicalization
- Assess cross-domain performance



Rule-based lexicalization strategy

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.