Towards automatically extracting morphosyntactical error patterns from L1-L2 parallel dependency treebanks

18th Workshop on Innovative Use of NLP for Building Educational Applications

Arianna Masciolini, Elena Volodina and Dana Dannélls

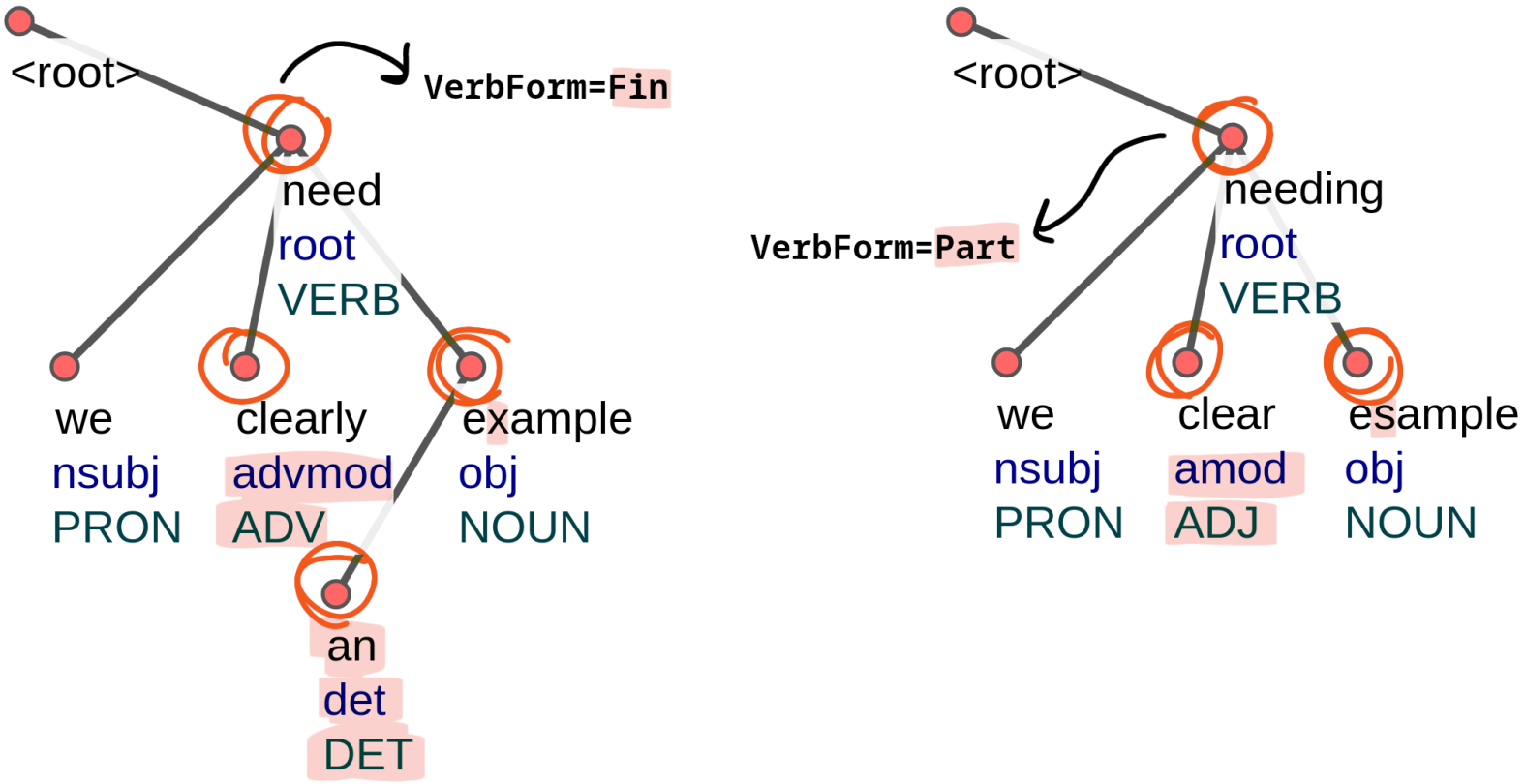
Språkbanken Text, University of Gothenburg

13.07.2023

L1-L2 treebanks

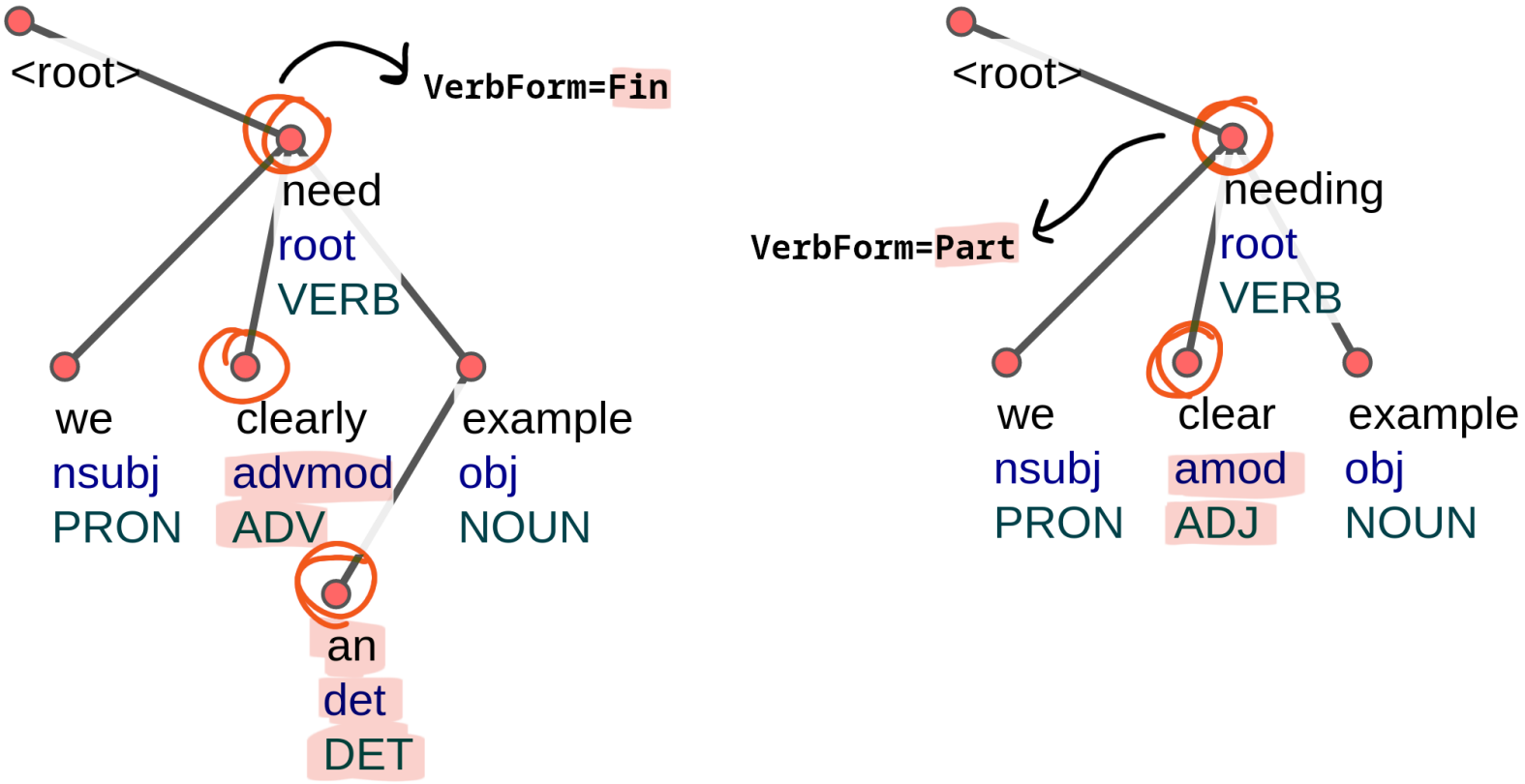
- learner sentences ‖ correction hypotheses
- no explicit error labelling,
  just morphosyntactical annotation
- main design goal: interoperability → UD

Example



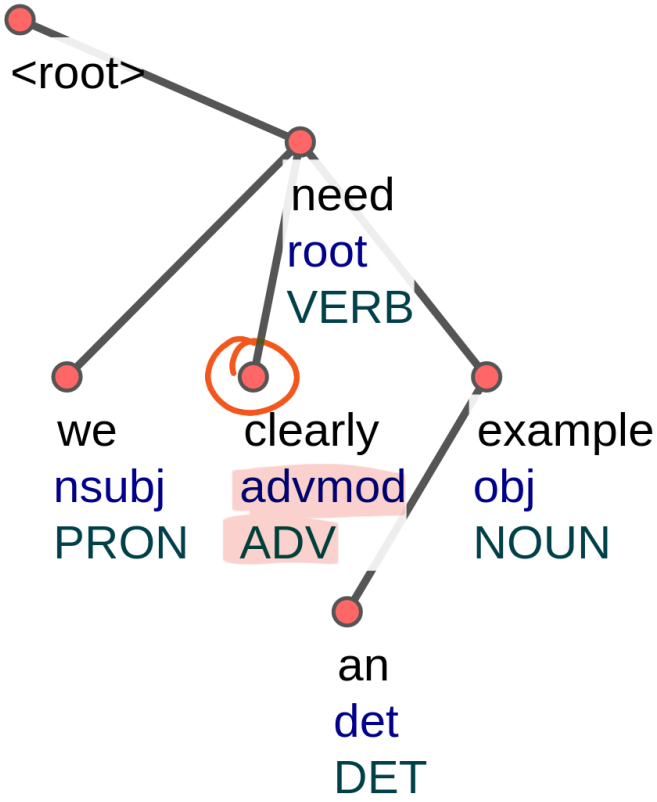
⟨we clear*ly* need~~ing~~ *an* e**x**ample, we clear needing _ esample⟩

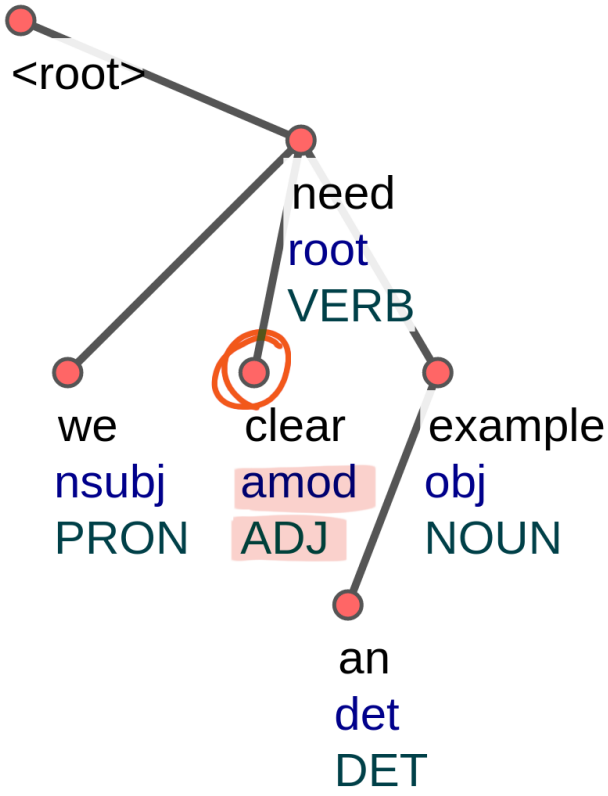
Example



⟨we clear*ly* need~~ing~~ *an* example, we clear needing _ example⟩

Example





⟨we clear*ly* need an example, we clear need an example⟩

Tasks

- find instances of specific error patterns → L2-UD query engine[1]
- automatically classify syntactical errors → SErCL[2]

[1] Masciolini, 2023
[2] Choshen et al., 2020

Tasks

- find instances of specific error patterns → L2-UD query engine[1]
- automatically classify syntactical errors → SErCL[2]
- **extract machine-readable error patterns**

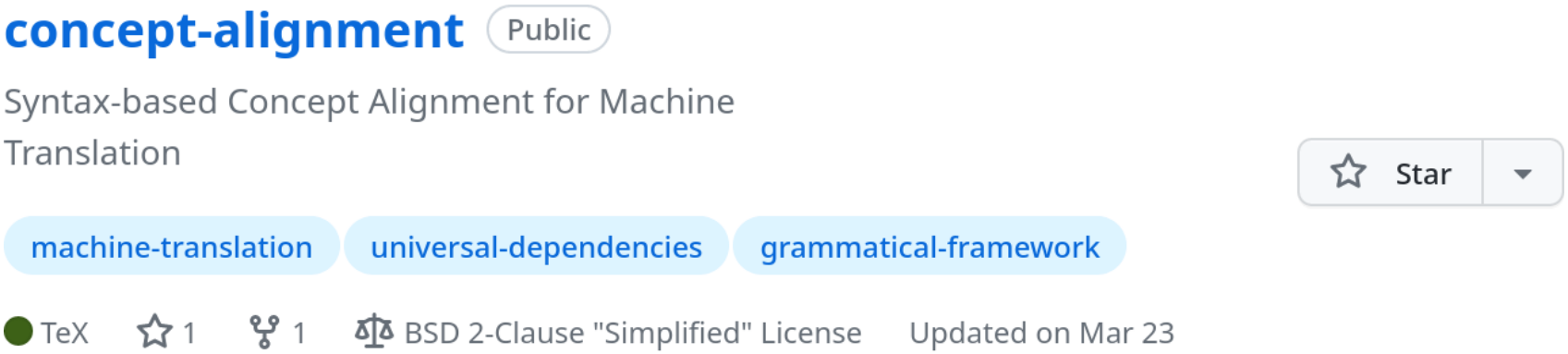
[1] Masciolini, 2023
[2] Choshen et al., 2020

Error pattern extraction

1. **error detection**: align L1-L2 sentences and filter discrepant alignment
2. **pattern generation**: convert pairs of UD subtrees into machine-readable error patterns

Step 1: error detection

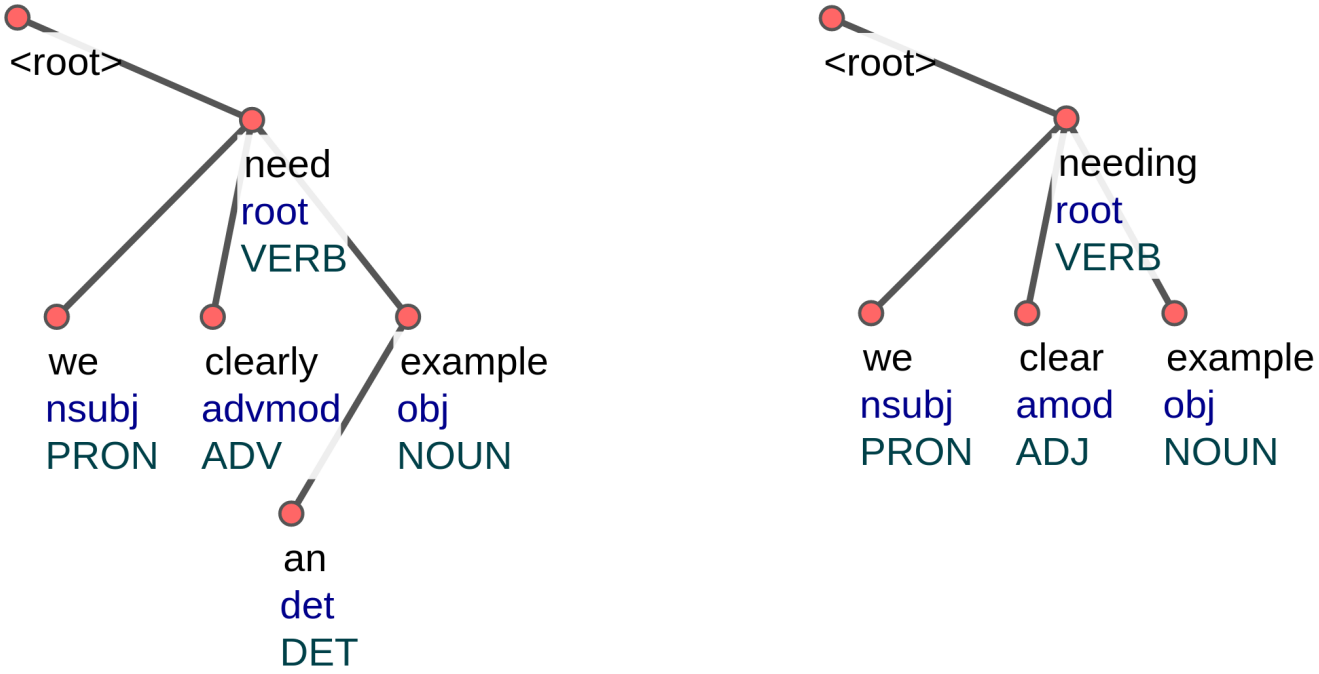
concept-alignment



`github.com/harisont/concept-alignment`

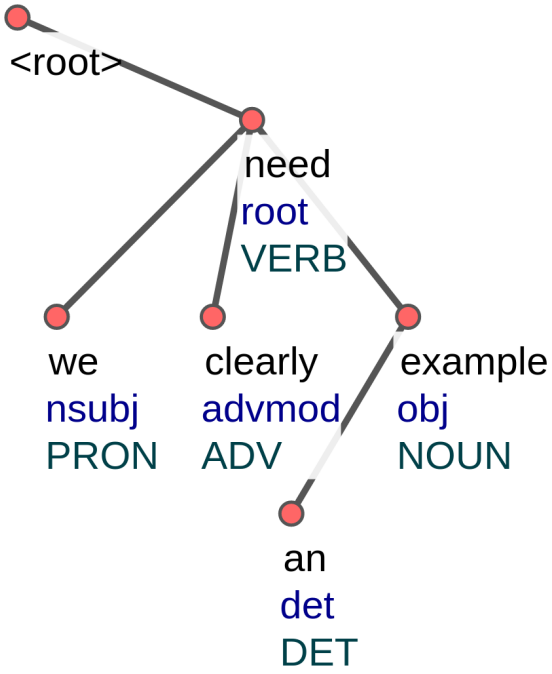
- extracts **subtree alignments** from parallel UD treebanks
- syntax-based but language-agnostic
- designed to generate translation lexica, but easy to adapt to the L1-L2 case

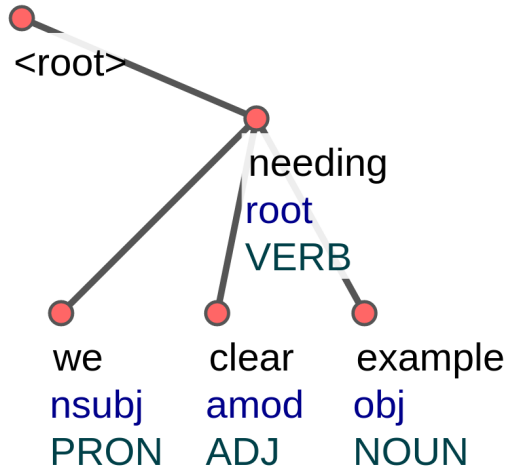
Alignments



- ⟨we clearly need an example, we clear needing example⟩, ⟨need, needing⟩
- ⟨we, we⟩
- ⟨clearly, clear⟩
- ⟨an example, example⟩, ⟨example, example⟩

Errors





- ⟨we **clearly need an example**, we **clear needing example**⟩*, ⟨**need**, **needing**⟩*
- ⟨we, we⟩
- ⟨**clearly**, **clear**⟩*
- ⟨**an example**, **example**⟩*, ⟨example, example⟩

Step 2: pattern generation

Query languages for UD trees

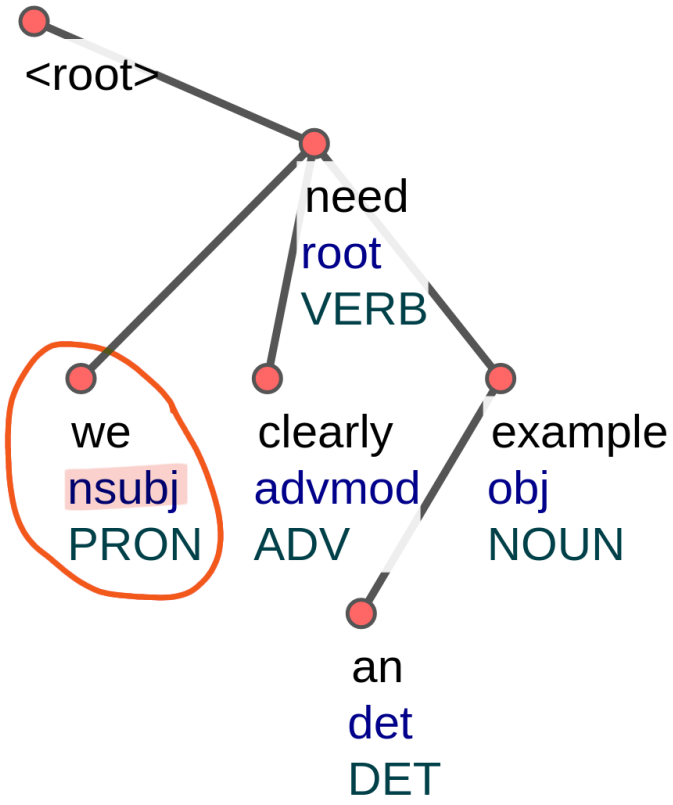
- several options to choose from
  - PML-TQ, Grew-match, UDAPI...
- decided on `gf-ud`'s embedded query language
  - sufficiently expressive and user-friendly
  - easy to use as a library

UD patterns in `gf-ud`

pattern type	**example**
single-token patterns	`DEPREL "nsubj"`
tree patterns	`TREE (POS "NOUN") [DEPREL "det"]`
sequence patterns	`SEQUENCE [DEPREL "advmod", POS "VERB"]`
logical operators[3]	`OR [POS "NOUN", POS "PRON"]`

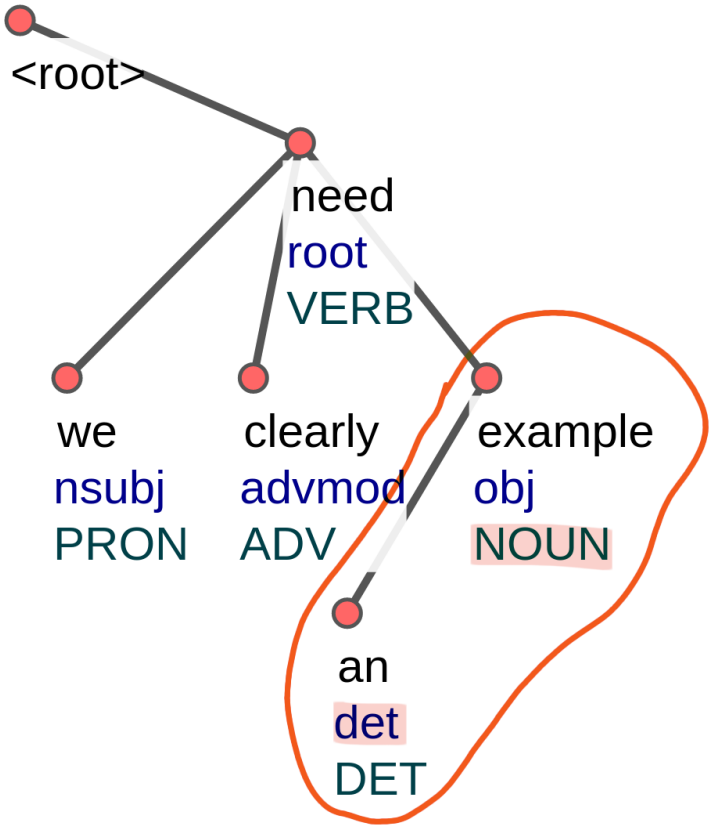
[3] AND, OR, NOT

Single-token patterns



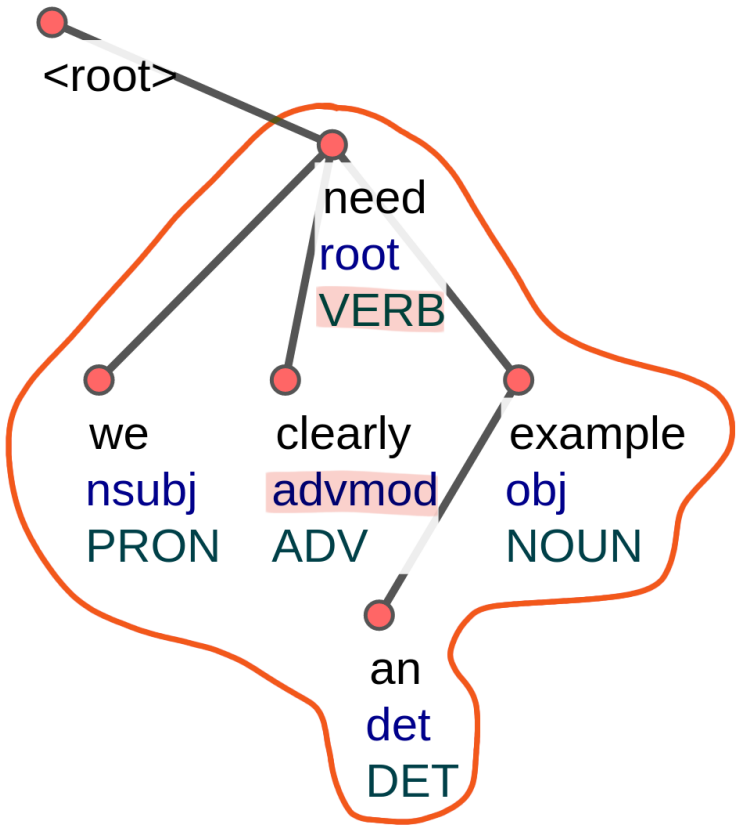

```
DEPREL "nsubj"
```

Tree patterns



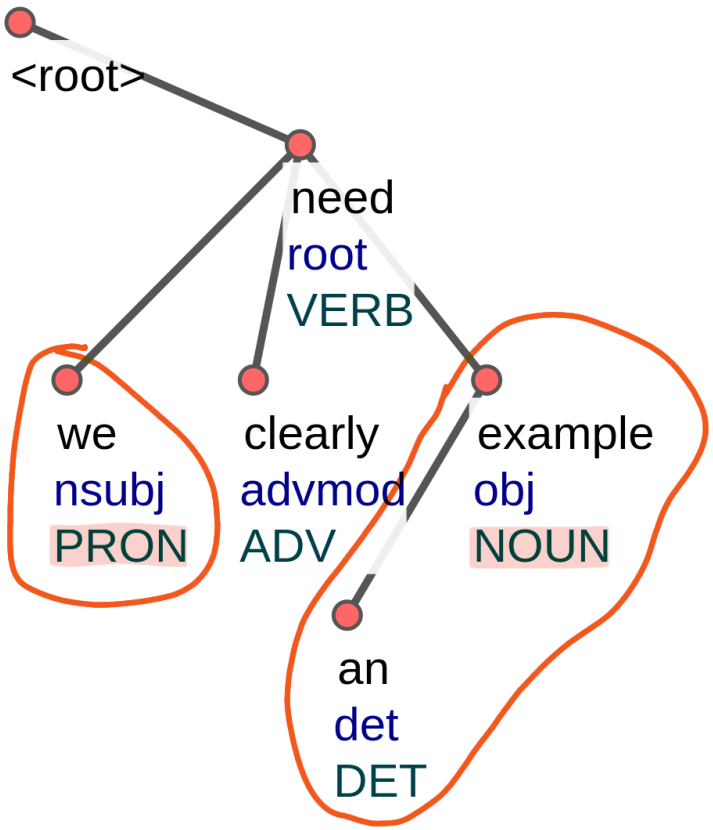

```
TREE (POS "NOUN") [DEPREL "det"]
```

Sequence patterns



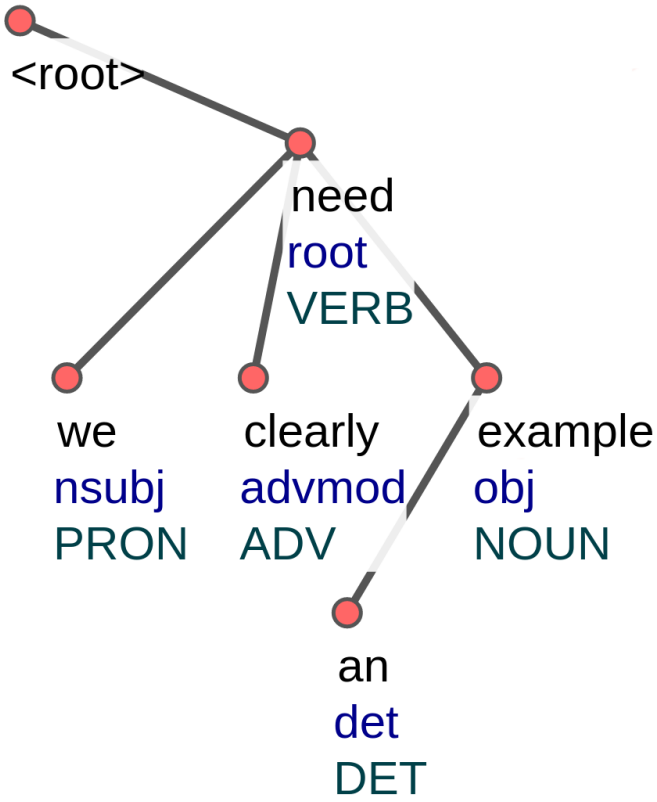

```
SEQUENCE [DEPREL "advmod", POS "VERB"]
```

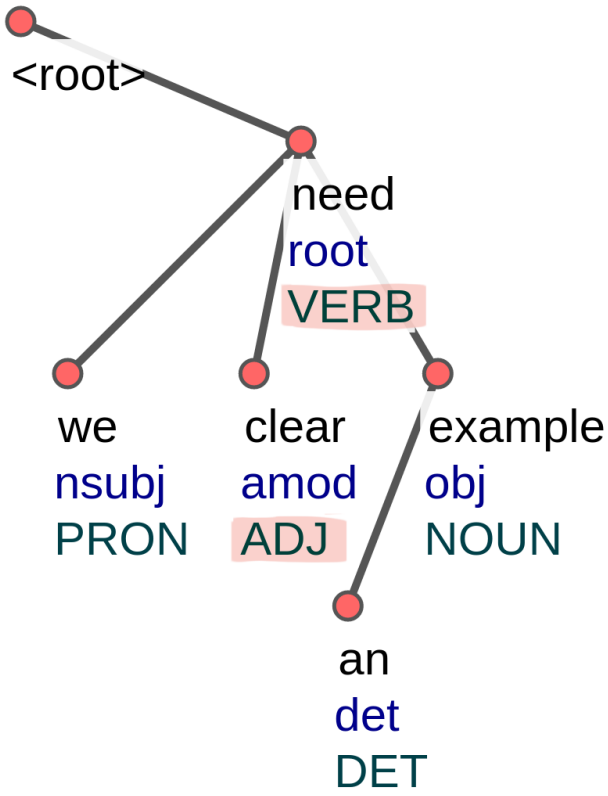
Logical operators




```
OR [POS "NOUN", POS "PRON"]
```

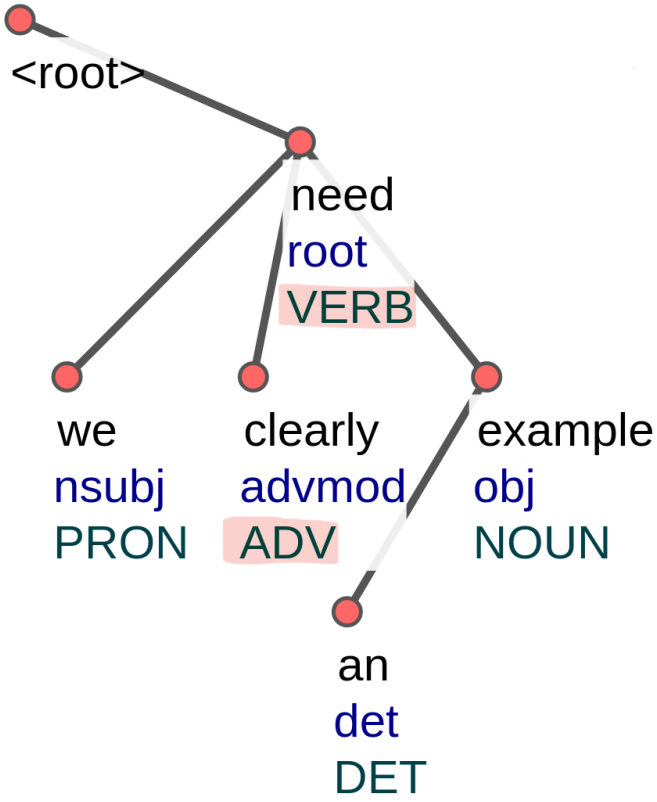
L1-L2 UD patterns

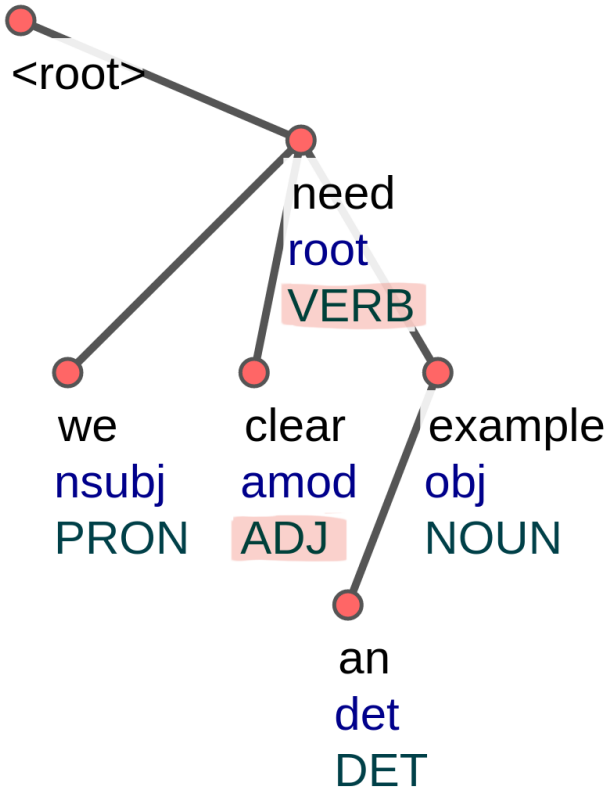





```
TREE_ (POS "VERB") [POS "ADJ"]
```

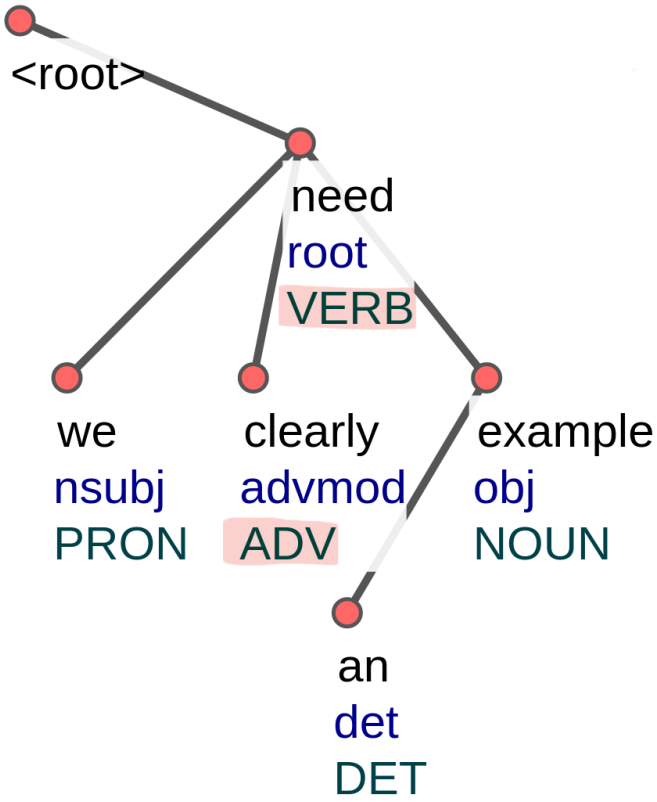
L1-L2 UD patterns

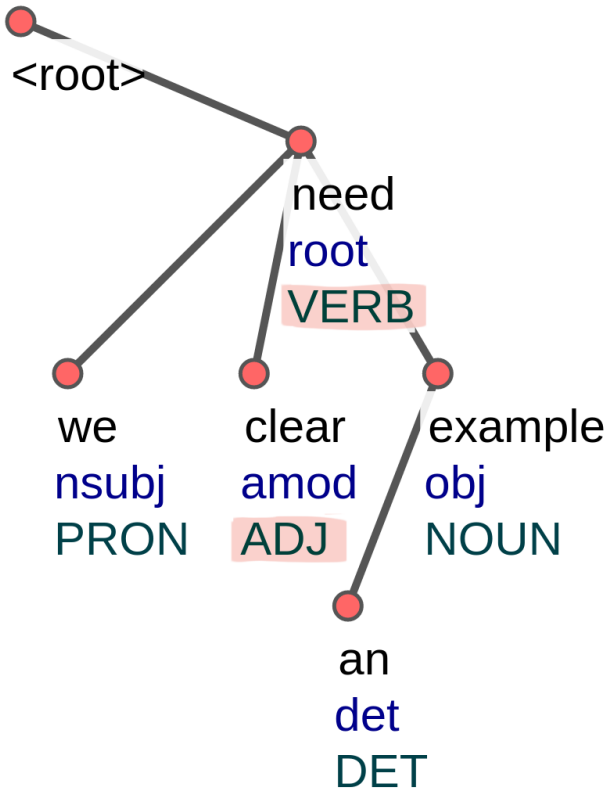





```
⟨TREE_ (POS "VERB") [POS "ADV"], TREE_ (POS "VERB") [POS "ADJ"]⟩
```

L1-L2 UD patterns






```
TREE_ (POS "VERB") [POS "{ADV -> ADJ}"]
```

Simplification strategies

0. Automatically generated L1-L2 pattern

```
TREE
   (AND [FORM "need", LEMMA "need", POS "VERB", XPOS "VBP", DEPREL "root",
        FEATS "Mood=Ind|Number=Plur|Person=1|Tense=Pres|VerbForm=Fin"])
   [AND [FORM "we", LEMMA "we", POS "PRON", XPOS "PRP", DEPREL "nsubj"
        FEATS "Case=Nom|Number=Plur|Person=1|PronType=Prs"],
    AND [FORM {"clearly" -> "clear"}, LEMMA {"clearly" -> "clear"},
        POS {"ADV" -> "ADJ"}, XPOS {"RB" -> "JJ"},
        FEATS {"_" -> "Degree=Pos"}, DEPREL {"advmod" -> "amod"}],
    TREE
      (AND [FORM "example", LEMMA "example", POS "NOUN",
           XPOS "NN", FEATS "Number=Sing", DEPREL "obj"])
      [AND [FORM "an", LEMMA "a", POS "DET", XPOS "DT",
           FEATS "Definite=Ind|PronType=Art", DEPREL "det"]
      ]
   ]
```

Simplification strategies

1. Filtering by CoNNL-U field

Ignoring FORM, LEMMA, XPOS and DEPREL:

```
TREE
   (AND [POS "VERB",
         FEATS "Mood=Ind|Number=Plur|Person=1|Tense=Pres|VerbForm=Fin"])
   [AND [POS "PRON", FEATS "Case=Nom|Number=Plur|Person=1|PronType=Prs"],
    AND [POS {"ADV" -> "ADJ"}, FEATS {"_" -> "Degree=Pos"}],
    TREE
      (AND [POS "NOUN", FEATS "Number=Sing"])
      [AND [POS "DET", FEATS "Definite=Ind|PronType=Art"]]
   ]
```

Simplification strategies

2. Removal of never-discrepant fields

In all alignments, FEATS is either identical both in the L1 and in the L2 or absent in one of the components:

```
TREE
   (AND [POS "VERB"])
   [AND [POS "PRON"],
    AND [POS {"ADV" -> "ADJ"}],
    TREE (AND [POS "NOUN"]) [AND [POS "DET"]]
   ]
```

Simplification strategies

3. Elimination of identical subpatterns

Removing identical subtrees[4]:

```
TREE_
    (AND [POS "VERB"])
    [AND [POS {"ADV" -> "ADJ"}]]
```

[4] optionally, identical roots can be replaced with the wildcard pattern `TRUE` too

Simplification strategies

4. Monolingual single-pattern simplifications

`AND [p]` is equivalent to `p`:

```
TREE_ (POS "VERB") [POS {"ADV" -> "ADJ"}]
```

Preliminary evaluation

Experimental design

- evaluation through a **similar example retrieval task**:
  1. **extract** L1-L2 patterns from an error-correction input pair
  2. **query** an L1-L2 treebank with the extracted patterns
- interactive version available as prototype CALL application

Data

2 **datasets for linguistic acceptability judgments**:

- only one error per sentence
- filtered: only morphosyntactical errors
- automatically parsed with UDPipe 2

name	**language**	**size**[5]	**description**
BLiMP	English	14 996	artificially generated sentences
DaLAJ	Swedish	1 198	postprocessed L2 learner sentences

[5] post-filtering

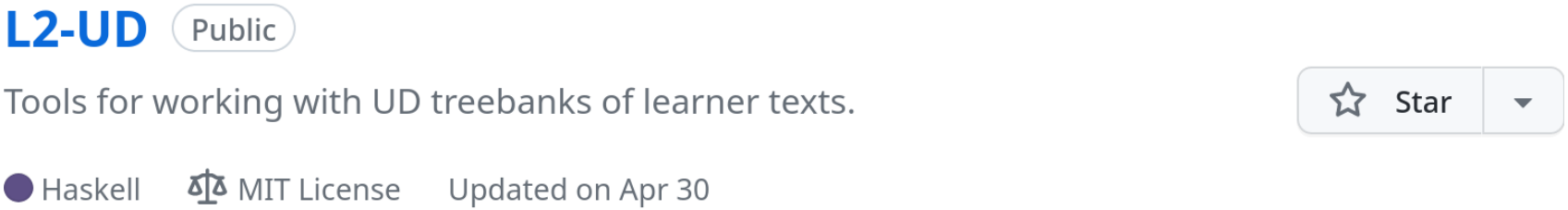
Results

	BLiMP	DaLAJ
R[6]	82%	69%
R_+[7]	82%	63%

[6] retrieval rate

[7] successful retrieval rate

Summarizing



`github.com/harisont/l2-ud`

- novel approach to error pattern extraction
- preliminary, bilingual evaluation on LA datasets giving promising results
- interactive similar example retrieval pipeline available as prototype CALL application

Future work

- extraction method:
  - handle nonexistent word forms
  - deal with real-world L2 data:
    - non-morphosyntactical errors (spelling, lexical. . .)
    - multiple overlapping errors
- example retrieval application:
  - implement pattern selection/ranking
  - build UI
- use L1-L2 patterns from feedback comment generation
- improve automatic annotation of L2 sentences

Thank you!

asynchronous questions/comments: arianna.masciolini@gu.se

References

- Leshem Choshen, Dmitry Nikolaev, Yevgeni Berzak, and Omri Abend. *Classifying syntactic errors in learner language*. In Proceedings of the 24th Conference on Computational Natural Language Learning, pages 97–107, Online, 2020. Association for Computational Linguistics
- Prasanth Kolachina and Aarnte Ranta. *From abstract syntax to Universal Dependencies*. Linguistic Issues in Language Technology, 13, 2016
- John Lee, Keying Li, and Herman Leung. *L1-L2 parallel dependency treebank as learner corpus*. In Proceedings of the 15th International Conference on Parsing Technologies, pages 44–49, Pisa, Italy, 2017. Association for Computational Linguistics

References

- Arianna Masciolini. *A query engine for L1-L2 parallel dependency treebanks*. In Proceedings of the 24th Nordic Conference on Computational Linguistics (NoDaLiDa), pages 574–587, Tórshavn, Faroe Islands, 2023. University of Tartu Library
- Arianna Masciolini and Aarne Ranta. *Grammar-based concept alignment for domain-specific Machine Translation*. In Proceedings of the Seventh International Workshop on Controlled Natural Language (CNL 2020/21), Amsterdam, Netherlands, 2021. Special Interest Group on Controlled Natural Language
- Aarne Ranta and Prasanth Kolachina. *From Universal Dependencies to abstract syntax*. In Proceedings of the NoDaLiDa 2017 Workshop on Universal Dependencies (UDW 2017), pages 107–116, Gothenburg, Sweden, 2017. Association for Computational Linguistics

References

- Elena Volodina, Yousuf Ali Mohammed, and Julia Klezl. *DaLAJ-a dataset for linguistic acceptability judgments for Swedish: Format, baseline, sharing*. arXiv preprint arXiv:2105.06681, 2021
- Alex Warstadt, Alicia Parrish, Haokun Liu, Anhad Mohananey, Wei Peng, Sheng-Fu Wang, and Samuel R. Bowman. *BLiMP: The benchmark of linguistic minimal pairs for English*. Transactions of the Association for Computational Linguistics, 8:377–392, 2020