

Applications of UD analysis

**syntactic queries, cross-linguistic
comparisons and language learning**

Arianna Masciolini and Aarne Ranta
LT2214 Computational Syntax

In today's lecture



1. some deptreepy magic
2. parallel syntactic queries with STuND
3. language learning applications

Deptreepy

Tree queries



usage:



```
python3 deptree.py <command> <arg>*
```

The command-arg combinations are

```
cosine_similarity <file> <file> <field>* # cosine similarity of treeban  
'match_trees <pattern>' # match entire trees  
'match_subtrees <pattern>' # match entire trees and recursively t  
'match_wordlines <pattern>' # match individual wordlines in all tr  
'match_segments <pattern>' # match contiguous, disjoint segments  
'change_wordlines <pattern>' # make changes in wordlines  
'change_subtrees <pattern>' # change subtrees recursively  
'statistics <field>*' # frequency-ordered statistics of <fie  
'treetype_statistics <field>*' # frequency-ordered statistics of type  
'head_dep_statistics <field>*' # frequency-ordered statistics of head  
'count_wordlines' # the number of wordlines
```

Compound tenses



- ✚ present continuous (*I **am showing** you the first query*)



- present continuous (*I **am showing** you the first query*):

TREE_

(FORM *ing)

(AND (LEMMA be) (FEATS *Tense=Pres*))

or even better:

TREE_

(AND (FEATS *VerbForm=Part*) (FEATS *Tense=Pres*))

(AND (LEMMA be) (FEATS *Tense=Pres*))



- present continuous (*I **am showing** you the first query*):

TREE_

(FORM *ing)

(AND (LEMMA be) (FEATS *Tense=Pres*))

or even better:

TREE_

(AND (FEATS *VerbForm=Part*) (FEATS *Tense=Pres*))

(AND (LEMMA be) (FEATS *Tense=Pres*))

- present perfect (*I **have just shown** you the first query*): ?



- present continuous (*I **am showing** you the first query*):

```
TREE_  
  (FORM *ing)  
  (AND (LEMMA be) (FEATS *Tense=Pres*))
```

or even better:

```
TREE_  
  (AND (FEATS *VerbForm=Part*) (FEATS *Tense=Pres*))  
  (AND (LEMMA be) (FEATS *Tense=Pres*))
```

- present perfect (*I **have just shown** you the first query*): ?
- ...

Márton András Tóth

Doctoral Student

Department of Swedish, multilingualism, language technology →

MAIL

marton.toth@gu.se

VISITING ADDRESS

Renströmsgatan 6
41255 Göteborg

POSTAL ADDRESS

Box 200
40530 Göteborg



About Márton András Tóth

I am a PhD student in Nordic Languages with main research focus on grammar. My PhD thesis concerns the use of the present perfect in Swedish (e.g. jag har ätit "I have eaten"). I am mostly interested in what characteristic features Swedish has in its tense-aspect system compared to other languages.



STUnD

Public

A GUI Search Tool for (bilingual) parallel Universal Dependencies treebanks

Haskell MIT License Updated on Jan 25

☆ Star

- ❏ Haskell prototype (small syntactic differences in the queries)
- ❏ available for download at github.com/harisont/STUnD

STUnD demo



1. sentences are aligned based on the UD analysis, resulting in a set of subtree correspondences¹

¹ how? See github.com/harisont/concept-alignment



1. sentences are aligned based on the UD analysis, resulting in a set of subtree correspondences¹
2. the query is rewritten as two separate patterns

¹ how? See github.com/harisont/concept-alignment



1. sentences are aligned based on the UD analysis, resulting in a set of subtree correspondences¹
2. the query is rewritten as two separate patterns
3. each query is matched against the correspondences found at step 1 (similar to deptreepy's `match_trees`, but in parallel)

¹ how? See github.com/harisont/concept-alignment



1. sentences are aligned based on the UD analysis, resulting in a set of subtree correspondences¹
2. the query is rewritten as two separate patterns
3. each query is matched against the correspondences found at step 1 (similar to deptreepy's `match_trees`, but in parallel)
4. if the user provided a replacement pattern, it is used to refine the search results

¹ how? See github.com/harisont/concept-alignment



Comparing:

- ✚ a text and its translation to another language



Comparing:

- ❑ a text and its translation to another language
- ❑ same text analyzed with two different parsers

Comparing:

- ❑ a text and its translation to another language
- ❑ same text analyzed with two different parsers
- ❑ **ungrammatical sentences vs. corrections hypotheses**

Yet another STUnD demo



- ❖ can we also infer error patterns from L1-L2 treebanks?
- ❖ and possibly: can we **generate feedback comments** on the basis of error patterns?



Given a learner sentence:



Given a learner sentence:

1. obtain correction hypothesis



Given a learner sentence:

1. obtain correction hypothesis
2. annotate learner sentence and correction in UD



Given a learner sentence:

1. obtain correction hypothesis
2. annotate learner sentence and correction in UD
3. extract error patterns



Given a learner sentence:

1. obtain correction hypothesis
2. annotate learner sentence and correction in UD
3. extract error patterns
4. generate feedback comments with GF

1. Grammatical Error Correction



“detta mening korrekt grammatisk?”



“Är denna mening grammatiskt korrekt?”

~“these sentence correct grammatical?” → “Is this sentence grammatically correct?”

1. Grammatical Error Correction



- ❑ well-established NLP task
- ❑ often solvable via back-and-forth translation

Back-and-forth translation



Swedish (detected) ↕ English (UK) Glossary

detta mening korrekt grammatisk? × Is this sentence grammatically correct?

Alternatives:

- Is this sentence correctly grammatical?
- This sentence is correctly grammatical?
- this sentence correctly grammatically?

👍 👎 ✎ 📄 🔗

Back-and-forth translation



English ▾ ↔ Swedish ▾ Glossary

Is this sentence grammatically correct? ×

Är denna mening grammatiskt korrekt?

Alternatives:

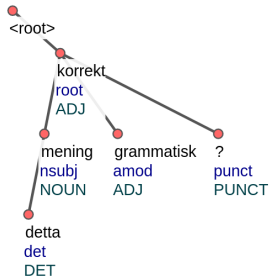
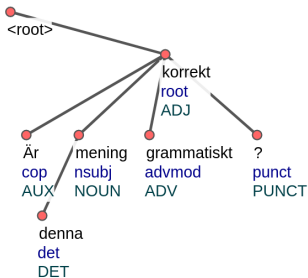
- Är meningen grammatiskt korrekt?
- Är den här meningen grammatiskt korrekt?
- Är denna mening grammatisk korrekt?

👍 👎 📄 🔗

2. UD annotation



⟨“Är denna mening grammatiskt korrekt?”, “detta mening korrekt grammatisk?”⟩

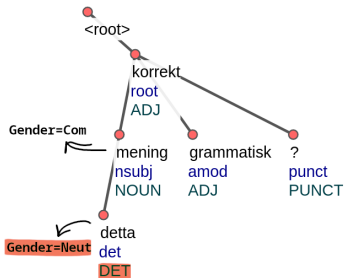
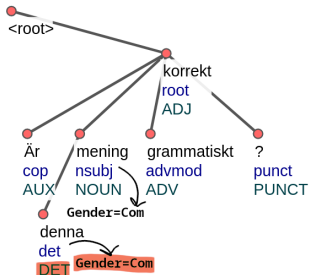


2. UD annotation



- ❖ existing UD parsers such as UDPipe perform well on standard language, L2 is way more challenging
- ❖ a task for ML (neural networks?)
- ❖ this is what I'm working on right now:
 - ❖ building a reference treebank
 - ❖ crazy experiments with using corrections as training data
 - ❖ less crazy experiments with synthetic errors

3. Error pattern extraction



AND [POS "DET", FEATS_ "Gender={Com -> Neutr}"], or
TREE_ (AND [POS "NOUN", FEATS_ "Gender=Com"]) [AND [POS
"DET", FEATS_ "Gender=Neutr"]]

3. Error pattern extraction



- ❖ creation of error patterns (“queries”) *from* UD tree pairs
- ❖ also something I’ve been working on, see (suboptimal) code at github.com/harison/L2-UD

4. Feedback comment generation



```
TREE (AND [POS "NOUN", FEATS_ "Gender=Com"])  
      [AND [POS "DET", FEATS_ "Gender=Neutr"]])
```



*The **determiner's** **gender is neutrum**, but the **gender** of the **noun** it refers to is **common**.*

4. Feedback comment generation



```
TREE (AND [POS "NOUN", FEATS_ "Gender=Com"])  
      [AND [POS "DET", FEATS_ "Gender=Neutr"]])
```



*OBS: detta **substantiv** är ett **en-ord**!*

4. Feedback comment generation



```
TREE (AND [POS "NOUN", FEATS_ "Gender=Com"])  
      [AND [POS "DET", FEATS_ "Gender=Neutr"]])
```



Pay attention to gender agreement!

4. Feedback comment generation



- ❖ (multilingual) data2text task
- ❖ Grammatical Framework is the perfect tool for the job!
- ❖ but this is a story for another day...



- ❖ good, consistent GEC?
- ❖ Swedish L2 treebanks?
- ❖ really good L2 parsers?
- ❖ improved pattern extraction?
- ❖ actual feedback comment generation?



But also:

- ❑ better subtree alignment?
- ❑ web version of deptreepy/STUnD?
- ❑ work on UD treebanks/parsers in general?
- ❑ dependency-based language learning applications? (see github.com/harison/advpy_nlp for inspiration)
- ❑ ...