UD-based analysis of grammatical errors in L2 texts

Arianna Masciolini

PROJECT DESCRIPTION

Grammar-based ICALL for self-study

Arianna Masciolini (`arianna.masciolini@gmail.com`)

June 23, 2022

Keywords

- **L2 grammar acquisition**

Keywords

- **L2 grammar acquisition**
- **tutorial ICALL**

Keywords

- L2 grammar acquisition
- tutorial ICALL
- exercise generation

Keywords

- **L2 grammar acquisition**
- **tutorial ICALL**
- **exercise generation**
- self-study $\rightarrow$ **automatic feedback**

Keywords

- **L2 grammar acquisition**
- **tutorial ICALL**
- **exercise generation**
- self-study $\rightarrow$ **automatic feedback**
- **multilingual**

Keywords

- **L2 grammar acquisition**
- **tutorial ICALL**
- **exercise generation**
- self-study → **automatic feedback**
- **multilingual**
- **grammar-based**

Keywords'

- **L2 grammar acquisition**
- ~~tutorial~~ **ICALL** (language tools)
- ~~exercise generation~~
- self-study → **automatic feedback** → **AWE/FCG**
- **multilingual**
- **grammar-based**, but also data-driven (**learner corpora**)

Why feedback?

MultiGED-2023

- "Multi" stands for "multilingual" here, but the original idea was for the task to be both
  - multilingual
  - multi-class (cf. Casademont Moner and Volodina, 2022)
- discussions about the data format

The "double CoNNL" format


```
# global.columns = ID FORM A D R TID

# sent_id = src:e1s1
1 Detta          _   _    M      1:2
2 mening         _   _           1:3
3 .              _   P̄    _      _

# sent_id = trg:e1s2
1 Korrekt        _   _    M̄      1:5
2 grammatisk     _   _    M̄|S    1:4
3 ?              _   _    _      1:6
```

```
# global.columns = ID FORM

# sent_id = trg:e1s1
1  Är
2  denna
3  mening
4  grammatiskt
5  korrekt
6  ?
```

The "double CoNNL" format

```
# global.columns = ID FORM A D R TID          # global.columns = ID FORM

# sent_id = src:e1s1                            # sent_id = trg:e1s1
1 Detta        _  _  M      1:2                 1 Är
2 mening       _  _         1:3                 2 denna
3 .            _  P  _      _                   3 mening
                                                4 grammatiskt
# sent_id = trg:e1s2                            5 korrekt
1 Korrekt      _  _         1:5                 6 ?
2 grammatisk   _  _  M|S    1:4
3 ?            _  _  _      1:6
```

- not used
- not necessarily a good idea

The "double CoNNL" format


```
# global.columns = ID FORM A D R TID        # global.columns = ID FORM

# sent_id = src:e1s1                         # sent_id = trg:e1s1
1 Detta          _  _   M      1:2           1 Är
2 mening         _  _   _      1:3           2 denna
3 .              _  P̄   _      _             3 mening
                                             4 grammatiskt
# sent_id = trg:e1s2                         5 korrekt
1 Korrekt        _  _   _      1:5           6 ?
2 grammatisk     _  _   M̄|S    1:4
3 ?              _  _   _      1:6
```

❖ not used

❖ not necessarily a good idea

but. . .

L1-L2 treebanks

L1-L2 Parallel Dependency Treebank as Learner Corpus

John Lee, Keying Li, Herman Leung
Department of Linguistics and Translation
City University of Hong Kong
jsylee@cityu.edu.hk, keyingli3-c@my.cityu.edu.hk, leung.hm@gmail.com

- L2 sentences // correction hypotheses
- no explicit error tagging, just **UD annotation**
  - better **interoperability** between learner corpora

Universal Dependencies 101
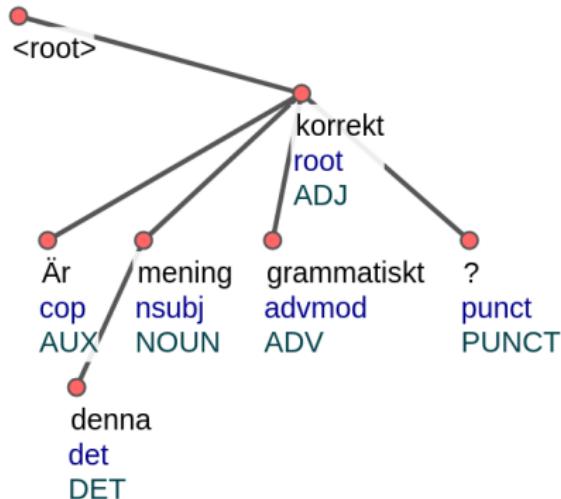
"Är denna mening grammatiskt korrekt?"

The CoNNL-U format

```
# text = Är denna mening grammatiskt korrekt?

1 Är vara AUX _ Mood=Ind|Tense=Pres|VerbForm=Fin|Voice=Act 5 cop _ _
2 denna denna DET _ Definite=Def|Gender=Com|Number=Sing|PronType=Dem 3 det _ _
3 mening mening NOUN _ Case=Nom|Definite=Ind|Gender=Com|Number=Sing 5 nsubj _ _
4 grammatiskt grammatisk ADV _ Degree=Pos 5 advmod _ _
5 korrekt korrekt ADJ _ Case=Nom|Definite=Ind|Degree=Pos|Gender=Com|Number=Sing 0 root _ _
6 ? ? PUNCT _ _ 5 punct _ _
```

ID · FORM · LEMMA · UPOS · FEATS · HEAD · DEPREL

The CoNNL-U format

```
# text = Är denna mening grammatiskt korrekt?

1 Är vara AUX _ Mood=Ind|Tense=Pres|VerbForm=Fin|Voice=Act 5 cop _ _
2 denna denna DET _ Definite=Def|Gender=Com|Number=Sing|PronType=Dem 3 det _ _
3 mening mening NOUN _ Case=Nom|Definite=Ind|Gender=Com|Number=Sing 5 nsubj _ _
4 grammatiskt grammatisk ADV _ Degree=Pos 5 advmod _ _
5 korrekt korrekt ADJ _ Case=Nom|Definite=Ind|Degree=Pos|Gender=Com|Number=Sing 0 root _ _
6 ? ? PUNCT _ _ 5 punct _ _
```

ID · FORM · LEMMA · UPOS · FEATS · HEAD · DEPREL

The CoNNL-U format

```
# text = Är denna mening grammatiskt korrekt?

1 Är vara AUX _ Mood=Ind|Tense=Pres|VerbForm=Fin|Voice=Act 5 cop _ _
2 denna denna DET _ Definite=Def|Gender=Com|Number=Sing|PronType=Dem 3 det _ _
3 mening mening NOUN _ Case=Nom|Definite=Ind|Gender=Com|Number=Sing 5 nsubj _ _
4 grammatiskt grammatisk ADV _ Degree=Pos 5 advmod _ _
5 korrekt korrekt ADJ _ Case=Nom|Definite=Ind|Degree=Pos|Gender=Com|Number=Sing 0 root _ _
6 ? ? PUNCT _ _ 5 punct _ _
```

ID · **FORM** · LEMMA · UPOS · FEATS · HEAD · DEPREL

The CoNNL-U format

```
# text = Är denna mening grammatiskt korrekt?

1 Är vara AUX _ Mood=Ind|Tense=Pres|VerbForm=Fin|Voice=Act 5 cop _ _
2 denna denna DET _ Definite=Def|Gender=Com|Number=Sing|PronType=Dem 3 det _ _
3 mening mening NOUN _ Case=Nom|Definite=Ind|Gender=Com|Number=Sing 5 nsubj _ _
4 grammatiskt grammatisk ADV _ Degree=Pos 5 advmod _ _
5 korrekt korrekt ADJ _ Case=Nom|Definite=Ind|Degree=Pos|Gender=Com|Number=Sing 0 root _ _
6 ? ? PUNCT _ _ 5 punct _ _
```

ID · FORM · **LEMMA** · UPOS · FEATS · HEAD · DEPREL

The CoNNL-U format

```
# text = Är denna mening grammatiskt korrekt?

1 Är vara AUX _ Mood=Ind|Tense=Pres|VerbForm=Fin|Voice=Act 5 cop _ _
2 denna denna DET _ Definite=Def|Gender=Com|Number=Sing|PronType=Dem 3 det _ _
3 mening mening NOUN _ Case=Nom|Definite=Ind|Gender=Com|Number=Sing 5 nsubj _ _
4 grammatiskt grammatisk ADV _ Degree=Pos 5 advmod _ _
5 korrekt korrekt ADJ _ Case=Nom|Definite=Ind|Degree=Pos|Gender=Com|Number=Sing 0 root _ _
6 ? ? PUNCT _ _ 5 punct _ _
```

ID · FORM · LEMMA · **UPOS** · FEATS · HEAD · DEPREL

The CoNNL-U format

```
# text = Är denna mening grammatiskt korrekt?

1 Är vara AUX _ Mood=Ind|Tense=Pres|VerbForm=Fin|Voice=Act 5 cop _ _
2 denna denna DET _ Definite=Def|Gender=Com|Number=Sing|PronType=Dem 3 det _ _
3 mening mening NOUN _ Case=Nom|Definite=Ind|Gender=Com|Number=Sing 5 nsubj _ _
4 grammatiskt grammatisk ADV _ Degree=Pos 5 advmod _ _
5 korrekt korrekt ADJ _ Case=Nom|Definite=Ind|Degree=Pos|Gender=Com|Number=Sing 0 root _ _
6 ? ? PUNCT _ _ 5 punct _ _
```

ID · FORM · LEMMA · UPOS · **FEATS** · HEAD · DEPREL

The CoNNL-U format

```
# text = Är denna mening grammatiskt korrekt?

1 Är vara AUX _ Mood=Ind|Tense=Pres|VerbForm=Fin|Voice=Act 5 cop _ _
2 denna denna DET _ Definite=Def|Gender=Com|Number=Sing|PronType=Dem 3 det _ _
3 mening mening NOUN _ Case=Nom|Definite=Ind|Gender=Com|Number=Sing 5 nsubj _ _
4 grammatiskt grammatisk ADV _ Degree=Pos 5 advmod _ _
5 korrekt korrekt ADJ _ Case=Nom|Definite=Ind|Degree=Pos|Gender=Com|Number=Sing 0 root _ _
6 ? ? PUNCT _ _ 5 punct _ _
```

ID · FORM · LEMMA · UPOS · FEATS · **HEAD** · DEPREL

The CoNNL-U format

```
# text = Är denna mening grammatiskt korrekt?

1 Är vara AUX _ Mood=Ind|Tense=Pres|VerbForm=Fin|Voice=Act 5 cop _ _
2 denna denna DET _ Definite=Def|Gender=Com|Number=Sing|PronType=Dem 3 det _ _
3 mening mening NOUN _ Case=Nom|Definite=Ind|Gender=Com|Number=Sing 5 nsubj _ _
4 grammatiskt grammatisk ADV _ Degree=Pos 5 advmod _ _
5 korrekt korrekt ADJ _ Case=Nom|Definite=Ind|Degree=Pos|Gender=Com|Number=Sing 0 root _ _
6 ? ? PUNCT _ _ 5 punct _ _
```

ID · FORM · LEMMA · UPOS · FEATS · HEAD · **DEPREL**

L1-L2 treebanks: example
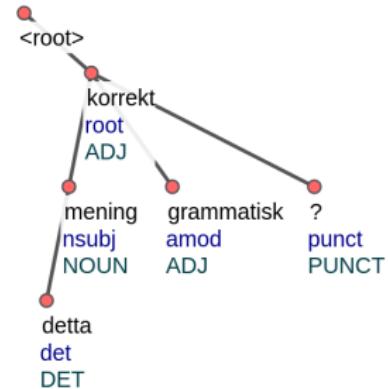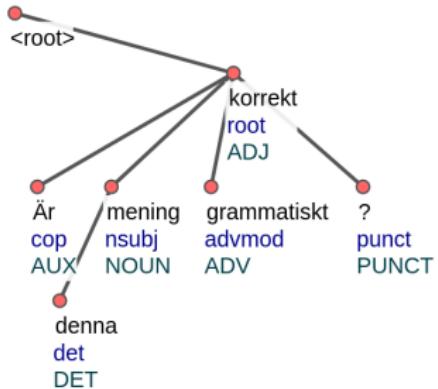
L1: "Är denna mening grammatiskt korrekt?" — L2: "detta mening korrekt grammatisk?"

L1-L2 treebanks: example

L1: "Är denna mening grammatiskt korrekt?" — L2: "detta mening korrekt grammatisk?"

L1-L2 treebanks: example
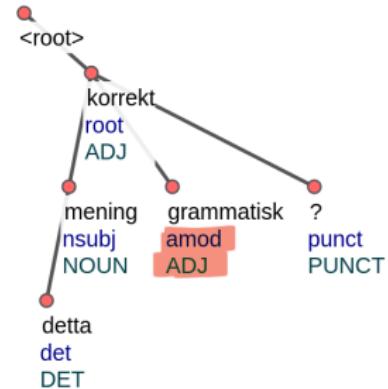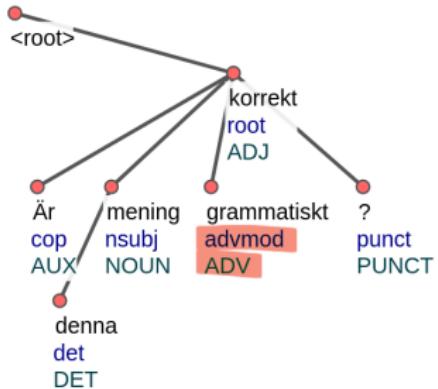
L1: "Är denna mening grammatiskt korrekt?" — L2: "detta mening korrekt grammatisk?"

L1-L2 treebanks: example
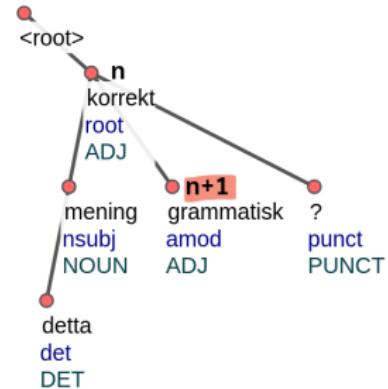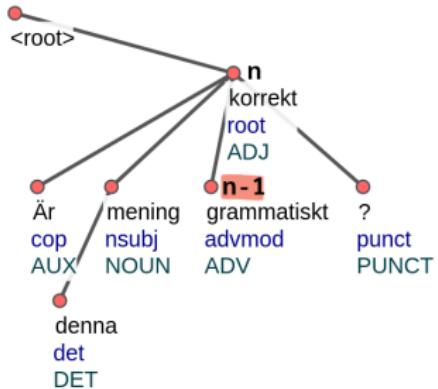
L1: "Är denna mening grammatiskt korrekt?" — L2: "detta mening korrekt grammatisk?"

# L1-L2 treebanks: example

L1: "Är denna mening grammatiskt korrekt?" — L2: "detta mening korrekt grammatisk?"

L1-L2 treebanks: example
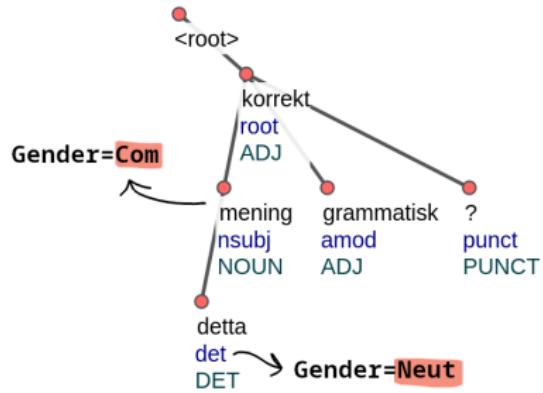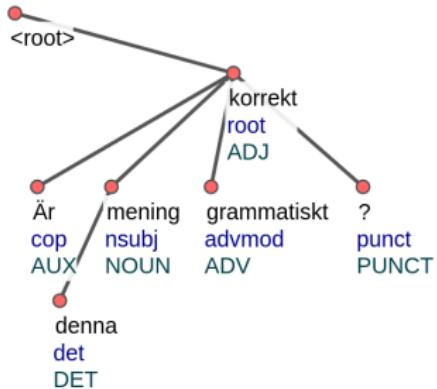
L1: "Är denna mening grammatiskt korrekt?" — L2: "detta mening korrekt grammatisk?"

Existing L1-L2 treebanks

treebank	language	n. sentences
TLE/ESL	English	5124
CFL	Chinese	451
VALICO-UD	Italian	398

L1-L2 treebanks and feedback

Key idea:

> *L1-L2 treebanks contain a lot of information useful for generating **feedback comments about morphosyntactic errors**.*

Steps

Given a learner sentence:

Steps

Given a learner sentence:

1. obtain correction hypothesis

Steps

Given a learner sentence:

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

Steps

Given a learner sentence:

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

Steps

Given a learner sentence:

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

Steps

Given a learner sentence:

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

1. Grammatical Error Correction

"detta mening korrekt grammatisk?"

↓

"Är denna mening grammatiskt korrekt?"

1. Grammatical Error Correction

- Well established task
- several promising approaches
  (see Bryant et al., 2022 for a recent survey)
- Swedish:
  - Granska system (Domeik et al., 2000)
  - Nyberg, 2022
  - Östling and Kurfali, 2022
- back-and-forth MT to the learner's L1 can help

1. Grammatical Error Correction

- Well established task
- several promising approaches
  (see Bryant et al., 2022 for a recent survey)
- Swedish:
  - Granska system (Domeik et al., 2000)
  - Nyberg, 2022
  - Östling and Kurfali, 2022
- back-and-forth MT to the learner's L1 can help

. . . not necessarily my problem

Steps

Given a learner sentence:

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

2. UD annotation

⟨"Är denna mening grammatiskt korrekt?", "detta mening korrekt grammatisk?"⟩

↓
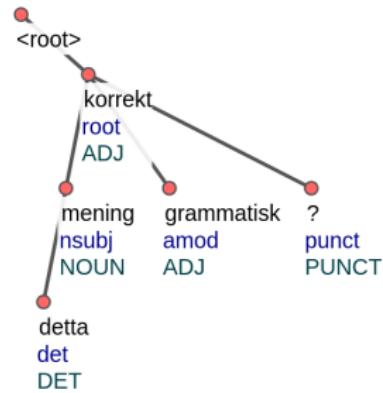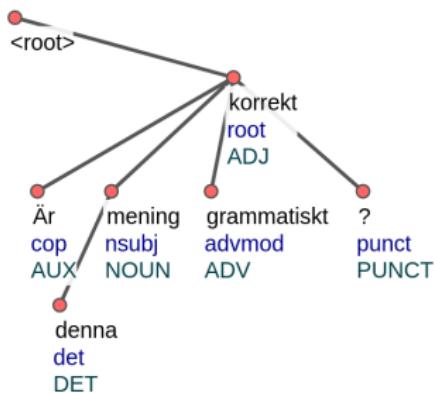
2. UD annotation

- Standard UD parsers perform well on L1 text, but **automatic annotation of L2 text remains challenging** [1]
- some tentative *ad-hoc* approaches:
  - rule-based error-diagnosing phase structure parser (Kakegawa et al., 2000)
  - ML-based error-repairing dependency parser (Sakaguchi and van Durme, 2017)

[1] Krivanek and Meurers; 2013; Huang et al., 2018; Volodina et al. 2022

2. UD annotation

. . . A problem for future me! Some (vague) ideas:

- just training a standard parser on a UD-annotated L2 corpus?
- L2 parsing "informed" by the L1 parse?
- in any case, it will be useful to have a Swedish L1-L2 treebank to use as a gold standard
   - starting in the near future, using SweLL data

Steps

Given a learner sentence:

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

3. Error pattern extraction

↓
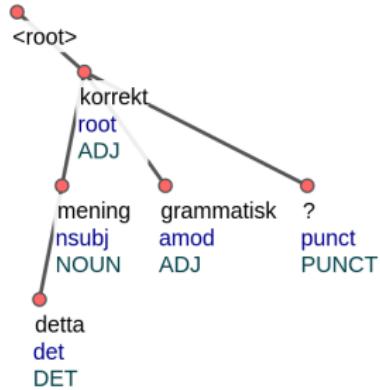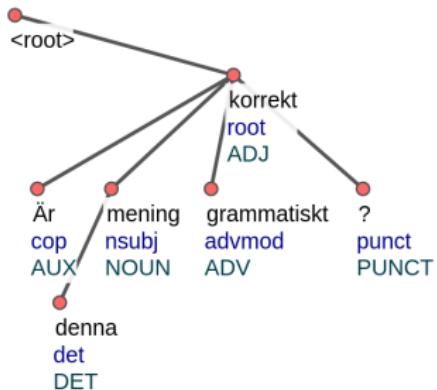
. . . some kind of machine-readable description of the errors?

3. Error pattern extraction

- new problem
- related to Choshen et al. (2020)'s work automatically inferring error classes from L1-L2 treebanks

3. Error pattern extraction

- new problem
- related to Choshen et al. (2020)'s work automatically inferring error classes from L1-L2 treebanks

. . . work in progress!

3. Error pattern extraction

Two subproblems:

1. locating error-correction pairs
   a. aligning the L2 sentence with its correction hypothesis
   b. selecting divergences due to morphosyntactical errors
2. representing them as machine-readable error patterns

3. Error pattern extraction

Two subproblems:

1. **locating error-correction pairs**
   a. aligning the L2 sentence with its correction hypothesis
   b. selecting divergences due to morphosyntactical errors
2. representing them as machine-readable error patterns

3.1 Error-correction pairs

concept-alignment (Public)

Syntax-based Concept Alignment for <u>Machine Translation</u> ?!

☆ Star ▾

`machine-translation` `universal-dependencies` `grammatical-framework`

● TeX   ☆ 1   ⑂ 1   ⚖ BSD 2-Clause "Simplified" License   Updated 2 weeks ago

- finds word- and phrase-level correspondences in parallel UD treebanks
- designed to build translation lexica, but fairly configurable
- the L1-L2 case is arguably easier than the multilingual one
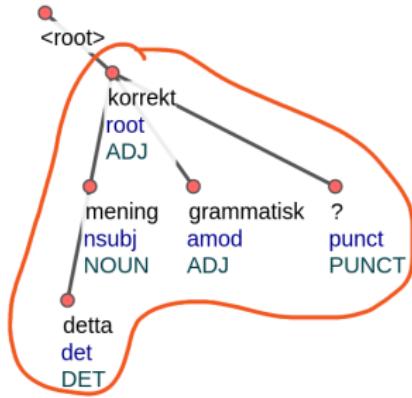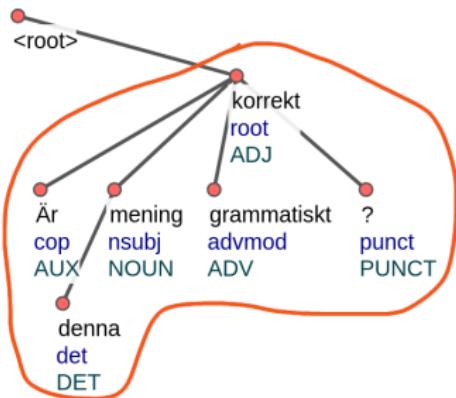
L1: "Är denna mening grammatiskt korrekt?" — L2: "detta mening korrekt grammatisk?"

3.1 Error-correction pairs
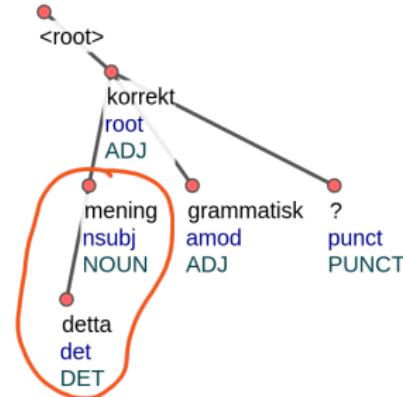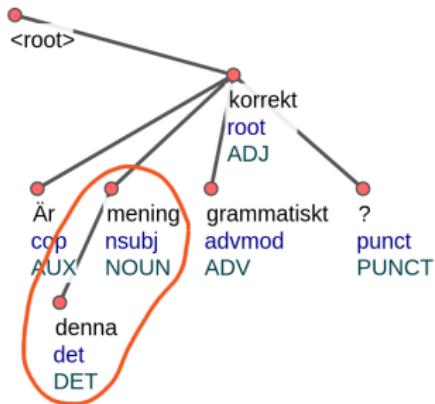
L1: "denna mening" — L2: "detta mening"

3.1 Error-correction pairs
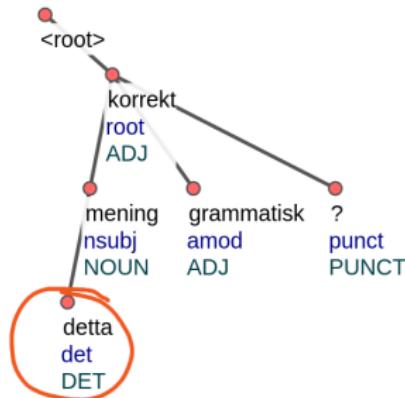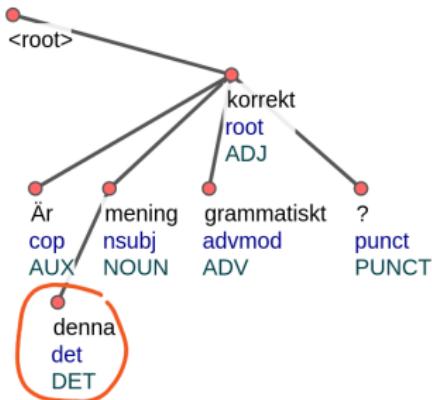
L1: "denna" — L2: "detta"

3.1 Error-correction pairs
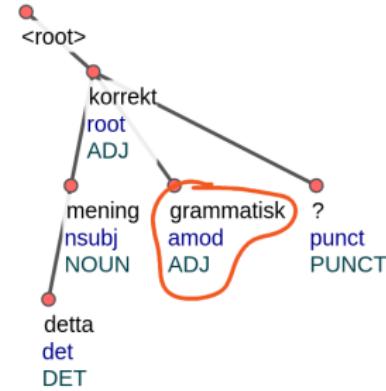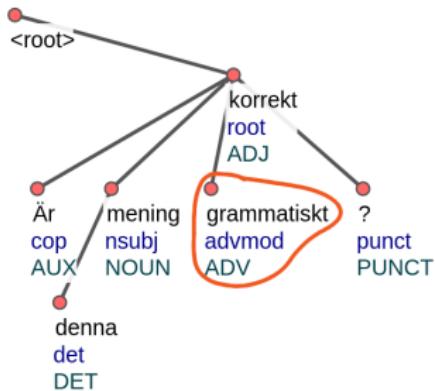
L1: "grammatiskt" — L2: "grammatisk"

3.1 Error-correction pairs
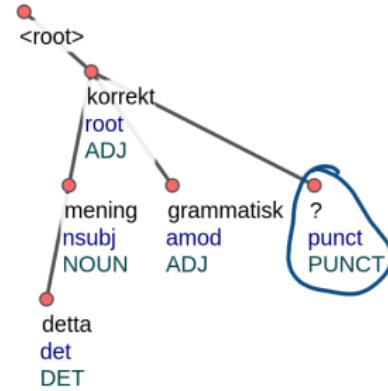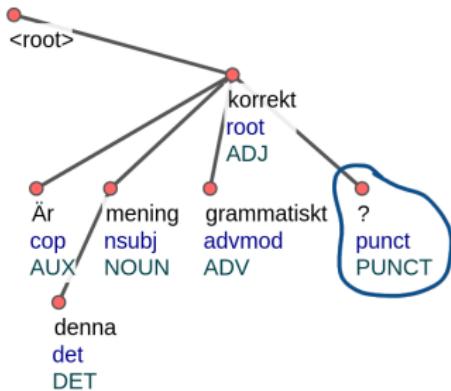
L1: "?" — L2: "?"

3.1 Error-correction pairs

- Does CA always work so well?
  - **no**

- Does CA always work so well?
  - **no**
- would it solve the problem completely if it did?
  - **not really**

- Does CA always work so well?
  - **no**
- would it solve the problem completely if it did?
  - **not really**
- does it help?
  - **yes!**

3. Error pattern extraction

Two subproblems:

1. locating error-correction pairs
   a. aligning the L2 sentence with its correction hypothesis
   b. selecting divergences due to morphosyntactical errors
2. **representing them as machine-readable error patterns**

3.2 Error patterns

How to represent error patterns?

- pairs of L1-L2 CoNNL-U subtrees
- using a query language for UD trees

3.2 Error patterns

How to represent error patterns?

- pairs of L1-L2 CoNNL-U subtrees
- **using a query language for UD trees**

Query languages for UD treebanks

- PML-TQ (Pajas and Štěpánek, 2009)
- TüNDRA (Martens, 2013)
- SETS (Luotolahti et al., 2015)
- Python (using UDAPI, Popel et al., 2017)
- Grew-match (Guillaume, 2021)
- . . .

Query languages for UD treebanks

gf-ud (Public)

Functions to analyse and manipulate dependency trees, as well as conversions between GF and dependency trees. The main use case is UD (Universal Dependencies), but the code is designed to be completely generic as for annotation scheme. This repository replaces the old gf-contrib/ud2gf code. It is also meant to be used in the 'vd' command of GF a...

● Grammatical Framework   ☆ 4   ⅄ 13   ⊙ 6   ⑂ 3
Updated on Jan 10

UD patterns in gf-ud

pattern type	example
single-token patterns	`POS "DET"`
tree patterns	`TREE (POS "NOUN") [DEPREL "det"]`
sequence patterns	`SEQUENCE [POS "DET", POS "NOUN"]`
logical operators	`AND [POS "NOUN", DEPREL "nsubj"]`

UD patterns in gf-ud

pattern type	example
single-token patterns	`POS "DET"`
tree patterns	`TREE (POS "NOUN") [DEPREL "det"]`
sequence patterns	`SEQUENCE [POS "DET", POS "NOUN"]`
logical operators	`AND [POS "NOUN", DEPREL "nsubj"]`

UD patterns in gf-ud

POS "DET"

UD patterns in gf-ud

pattern type	example
single-token patterns	`POS "DET"`
tree patterns	`TREE (POS "NOUN") [DEPREL "det"]`
sequence patterns	`SEQUENCE [POS "DET", POS "NOUN"]`
logical operators	`AND [POS "NOUN", DEPREL "nsubj"]`

UD patterns in gf-ud

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

UD patterns in gf-ud

pattern type	example
single-token patterns	`POS "DET"`
tree patterns	`TREE (POS "NOUN") [DEPREL "det"]`
sequence patterns	`SEQUENCE [POS "DET", POS "NOUN"]`
logical operators	`AND [POS "NOUN", DEPREL "nsubj"]`
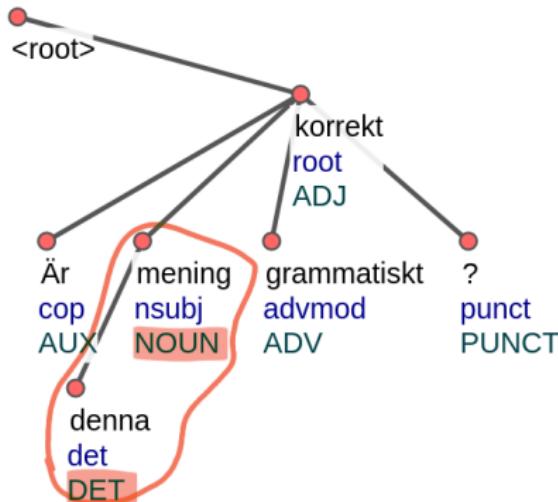
UD patterns in gf-ud

SEQUENCE [POS "DET", POS "NOUN"]

UD patterns in gf-ud

pattern type	example
single-token patterns	`POS "DET"`
tree patterns	`TREE (POS "NOUN") [DEPREL "det"]`
sequence patterns	`SEQUENCE [POS "DET", POS "NOUN"]`
logical operators	`AND [POS "NOUN", DEPREL "nsubj"]`

UD patterns in gf-ud

AND [POS "NOUN", DEPREL "nsubj"]
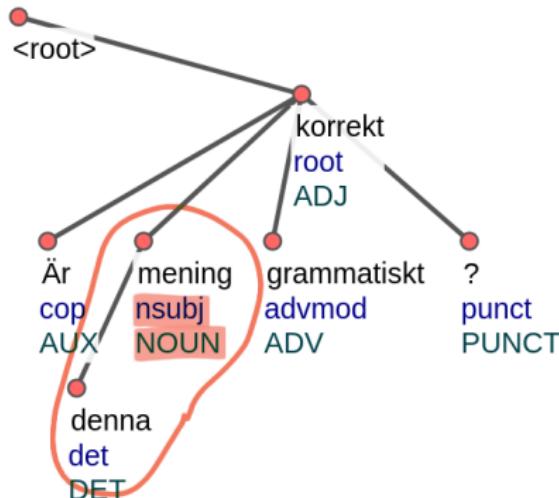
L1-L2 UD patterns

Many errors can be represented as UD patterns describing the L2
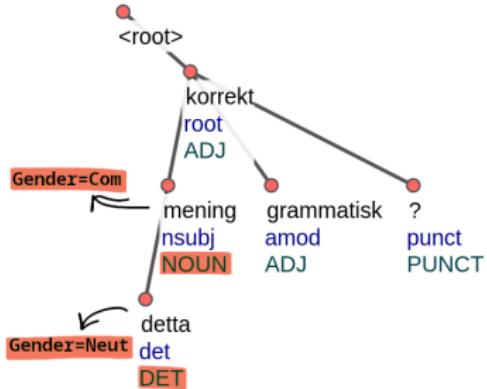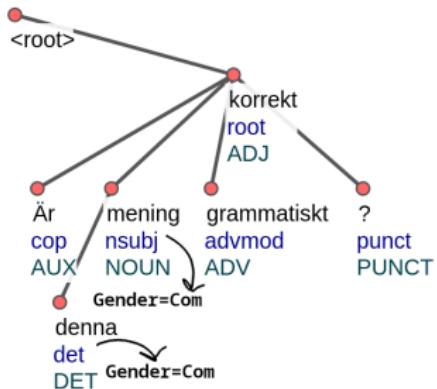
TREE (AND [POS "NOUN", FEATS_ "Gender=Com"]) [AND [POS "DET", FEATS_ "Gender=Neutr"]]

L1-L2 UD patterns

Sometimes, it is useful (or even necessary) to compare the L1 and L2 → **L1-L2 patterns** (pairs of UD patterns)
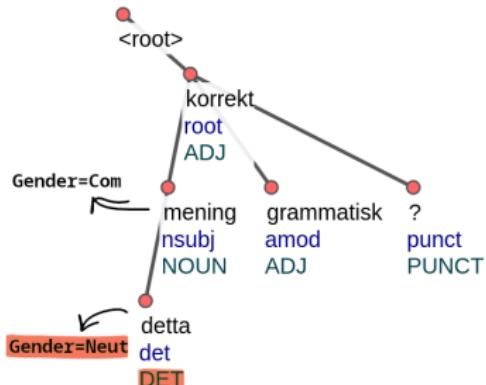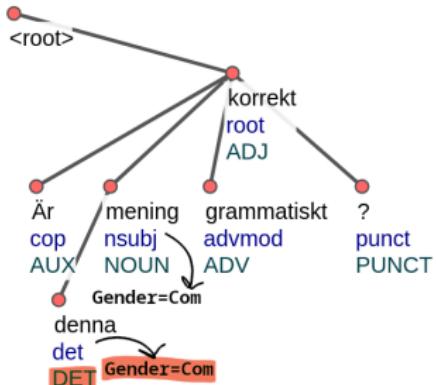
⟨ AND [POS "DET", FEATS_ "Gender=Com"], AND [POS "DET", FEATS_ "Gender=Neutr"] ⟩

3.2 Error patterns

Based on my ongoing work...

3.2 Error patterns

Based on my ongoing work...

- is this *the most* expressive query language out there?
    - **probably not**

3.2 Error patterns

Based on my ongoing work...

- is this *the most* expressive query language out there?
  - **probably not**
- is it expressive *enough*?
  - **yes!**

3.2 Error patterns

Based on my ongoing work. . .

- is this *the most* expressive query language out there?
  - **probably not**
- is it expressive *enough*?
  - **yes!**
- is it easy to integrate in my code, and to work with in general?
  - **very!**

3.2 Error patterns

Based on my ongoing work...

- is this *the most* expressive query language out there?
  - **probably not**
- is it expressive *enough*?
  - **yes!**
- is it easy to integrate in my code, and to work with in general?
  - **very!**

From a review:

> *The GF-UD query language seems user-friendly and expressive enough for a range of **queries** over UD treebanks.*

Queries!?

A query engine for L1-L2 parallel dependency treebanks

Arianna Masciolini

Published: 20 Mar 2023, Last Modified: 22 Mar 2023   NoDaLiDa 2023   Readers: 🌐 Everyone   Show Bibtex   Show Revisions

Keywords: L2 corpora, Universal Dependencies

TL;DR: Presentation of a query engine for retrieving errors from L1-L2 parallel dependency treebanks

Abstract: L1-L2 parallel dependency treebanks, aimed at achieving interoperability between learner corpora, consist of learner sentences paired with correction hypotheses. Rather than explicitly labelled for errors, sentences are morphosyntactically annotated following the Universal Dependencies standard, which enables error retrieval via tree queries. Work in this sense has, however, been limited. Addressing the limitations of the existing tools, we present a query engine for L1-L2 treebanks and evaluate it on a manually validated and an automatically parsed corpus.

Student Paper: Yes, the first author is a student

Add **Withdraw**

Reply Type: all ▾   Author: everybody ▾   Visible To: all readers ▾   Hidden From: nobody ▾   **3 Replies**

[-] **Paper Decision**

NoDaLiDa 2023 Conference Program Chairs

17 Mar 2023   NoDaLiDa 2023 Conference Paper102 Decision   Readers: 🌐 Everyone

Decision: Accept

Where is the code?

L2-UD (Public)
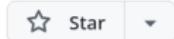
Tools for working with UD treebanks of learner texts.

⬤ Haskell   ⚖ MIT License   Updated 1 minute ago

☆ Star ▾

Contains both:

- the query engine
- the code for extracting error patterns (under development)

Steps

Given a learner sentence:

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

Feedback in CALL

"Är detta mening grammatiskt korrekt?"

type	example
correct/incorrect	Try again!
correct answer	Är denna mening grammatiskt korrekt?
highlighting	Är detta mening grammatiskt korrekt?
metalinguistic	Pay attention to gender agreement!
example	Detta är en exempelmening → Denna är en exempelmening
error label	M-Gend

Feedback in CALL

"Är detta mening grammatiskt korrekt?"

type	example
correct/incorrect	Try again!
correct answer	Är denna mening grammatiskt korrekt?
highlighting	Är detta mening grammatiskt korrekt?
metalinguistic	Pay attention to gender agreement!
example	Detta är en exempelmening → Denna är en exempelmening
error label	`M-Gend`

... or any combination of the above!

Feedback in CALL

"Är detta mening grammatiskt korrekt?"

type	example
correct/incorrect	Try again!
correct answer	Är denna mening grammatiskt korrekt?
highlighting	Är detta mening grammatiskt korrekt?
metalinguistic	Pay attention to gender agreement!
example	Detta är en exempelmening → Denna är en exempelmening
error label	`M-Gend`

. . . or any combination of the above!

Feedback in CALL

"Är detta mening grammatiskt korrekt?"

type	example
correct/incorrect	Try again!
correct answer	Är denna mening grammatiskt korrekt?
highlighting	Är detta mening grammatiskt korrekt?
metalinguistic	Pay attention to gender agreement!
example	Detta är en exempelmening → Denna är en exempelmening
error label	`M-Gend`

. . . or any combination of the above!

Are feedback comments useful?

Based on SLA research, **maybe**:

- Truscott, 1996 *convincedly* argues that grammar correction should be abandoned altogether...
- ...but Ferris, 1999's response article claims it does not do so *convincingly*...
- ...and the debate goes on...

Are feedback comments useful?

Some more useful questions:

- *what kind* of feedback is useful?
- in *which cases*?
- *how* should it be used?

Are feedback comments useful?

Some more useful questions:

- *what kind* of feedback is useful?
- in *which cases*?
- *how* should it be used?

. . . a flexible, general-purpose way to automatically generate feedback comments can be a tool to answer these questions!

Are feedback comments useful?

Some more useful questions:

- ***what kind* of feedback is useful?**
- in *which cases*?
- *how* should it be used?

. . . a flexible, general-purpose way to automatically generate feedback comments can be a tool to answer these questions!

What kind of feedback is useful?

Based on CALL research, metalinguistic feedback is **probably** useful:

- Heift, 2001 shows that student attend to metalinguistic feedback, even when they can request the correct answer
- Heift, 2004 suggests that metalinguistic feedback, combined with highlighting, has positive effects on learner uptake
- plenty of recent papers[1] and a shared task[2] on FCG

[1] Nagata et al., 2020; Hanawa et al, 2021; Huang et al., 2018; Galvan-Sosa et al., 2023...
[2] Nagata et al., 2021

4. Feedback Comment Generation

. . . (far) future work! Some (less vague) ideas:

- data2text task
  - error patterns → feedback comments, ideally:
    - in multiple languages
    - adjustable to the learner's level

↓

idea: **a GF CNL**

Grammatical Framework 101

A generative grammar formalism/programming language for **multilingual grammar engineering**:

- GF grammar = 1 *abstract syntax* + *n concrete syntaxes*
- especially well suited for defining *application grammars*
- interoperable with UD (does that help?)

FCG with GF

Parse error patterns, generate natural language sentences:

**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.*

FCG with GF

Parse error patterns, generate natural language sentences:

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

$\downarrow$

OBS: detta substantiv är ett en-ord!

FCG with GF

Parse error patterns, generate natural language sentences:

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

↓

*Pay attention to gender agreement!*

To summarize

Planned contributions:

- ☒ query engine for L1-L2 treebanks
- ☐ [WIP] error pattern extraction module
- ☐ [soon] L1-L2 Swedish treebank
- ☐ CNL for FCG
- ☐ some kind of demo application

In recent news. . .

Duolingo Explain My Answer
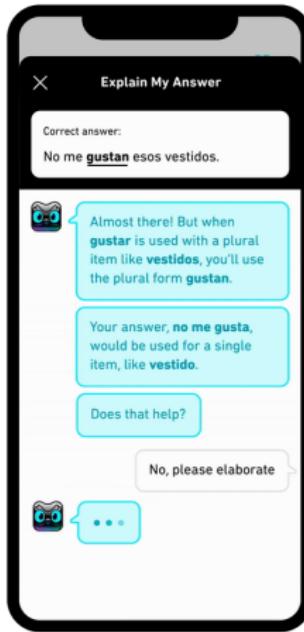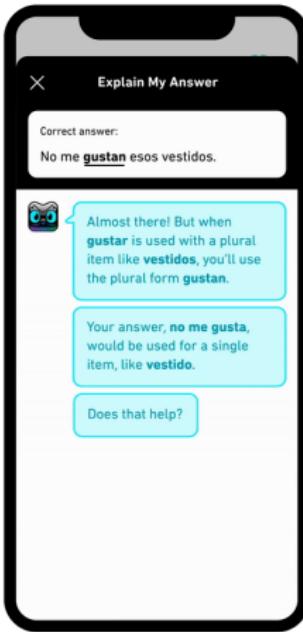
Thank you!

References

L1-L2 treebanks

- Yevgeni Berzak, Jessica Kenney, Carolyn Spadine, Jing Xian Wang, Lucia Lam, Keiko Sophie Mori, Sebastian Garza, and Boris Katz. *Universal Dependencies for learner English*. arXiv preprint `arXiv:1605.04278`, 2016

- Elisa Di Nuovo, Manuela Sanguinetti, Alessandro Mazzei, Elisa Corino, and Cristina Bosco. *VALICO-UD: Treebanking an Italian learner corpus in Universal Dependencies*. IJCoL. Italian Journal of Computational Linguistics, 8(8-1), 2022

- John SY Lee, Herman Leung, and Keying Li. *Towards Universal Dependencies for learner Chinese*. In Proceedings of the NoDaLiDa 2017 Workshop on Universal Dependencies (UDW 2017), pages 67–71, 2017

- John SY 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, 2017

References

Query tools and languages for UD treebanks

- Bruno Guillaume. *Graph matching and graph rewriting: GREW tools for corpus exploration, maintenance and conversion*. In Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations, pages 168–175, 2021

- Juhani Luotolahti, Jenna Kanerva, Sampo Pyysalo, and Filip Ginter. SETS: Scalable and Efficient Tree Search in dependency graphs. In Proceedings of the 2015 conference of the north american chapter of the association for computational linguistics: Demonstrations, pages 51–55, 2015

- Scott Martens. *TüNDRA: A web application for treebank search and visualization*. In The Twelfth Workshop on Treebanks and Linguistic Theories (TLT12), volume 133, 2013

- Petr Pajas and Jan Štěpánek. *System for querying syntactically annotated corpora*. In Proceedings of the ACL-IJCNLP 2009 Software Demonstrations, pages 33–36, 2009

- Martin Popel, Zdeněk Žabokrtský, and Martin Vojtek. *Udapi: Universal API for Universal Dependencies*. In Proceedings of the NoDaLiDa 2017 Workshop on Universal Dependencies (UDW 2017), pages 96–101, 2017

References

GEC

- Christopher Bryant, Zheng Yuan, Muhammad Reza Qorib, Hannan Cao, Hwee Tou Ng, and Ted Briscoe. *Grammatical Error Correction: A survey of the state of the art.* arXiv preprint `arXiv:2211.05166`, 2022

- Judith Casademont Moner and Elena Volodina. *Swedish MuClaGED: A new dataset for Grammatical Error Detection in Swedish.* In Proceedings of the 11th Workshop on NLP for Computer Assisted Language Learning, pages 36–45, 2022

- Rickard Domeij, Ola Knutsson, Johan Carlberger, and Viggo Kann. *Granska–an efficient hybrid system for Swedish grammar checking.* In Proceedings of the 12th Nordic Conference of Computational Linguistics (NODALIDA 1999), pages 49–56, 2000

- Martina Nyberg. *Grammatical Error Correction for Learners of Swedish as a Second Language.* Uppsala Universitet, 2022

- Robert Östling and Murathan Kurfali. *Really good Grammatical Error Correction, and how to evaluate it.* 2022

References

Automatic annotation of L2 texts

- Leshem Choshen, Dmitry Nikolaev, Yevgeni Berzak, and Omri Abend. *Classifying syntactic errors in learner language*. arXiv preprint `arXiv:2010.11032`, 2020

- Yan Huang, Akira Murakami, Theodora Alexopoulou, and Anna Korhonen. *Dependency parsing of learner English*. International Journal of Corpus Linguistics, 23(1):28–54, 2018

- Jun'ichi Kakegawa, Hisayuki Kanda, Eitaro Fujioka, Makoto Itami, and Kohji Itoh. *Diagnostic processing of Japanese for computer-assisted second language learning*. In Proceedings of the 38th Annual Meeting of the Association for Computational Linguistics, pages 537–546, 2000

- Julia Krivanek and Detmar Meurers. *Comparing rule-based and data-driven dependency parsing of learner language*. Computational dependency theory, 258:207, 2013

- Keisuke Sakaguchi, Matt Post, and Benjamin Van Durme. *Error-repair dependency parsing for ungrammatical texts*. In Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), pages 189–195, 2017

References

Elena Volodina, David Alfter, Therese Lindström Tiedemann, Maisa Susanna Lauriala, and Daniela Helena Piipponen. *Reliability of automatic linguistic annotation: native vs non-native texts.* In Selected papers from the CLARINAnnual Conference 2021. Linköping University Electronic Press (LiU E-Press), 2022

Feedback in SLA

Dana Ferris. *The case for grammar correction in L2 writing classes: A response to Truscott* (1996). Journal of second language writing, 8(1):1–11, 1999.

Dana R Ferris. The "grammar correction" debate in L2 writing: Where are we, and where do we go from here? (and what do we do in the meantime. . . ?). Journal of second language writing, 13(1):49–62, 2004

John Truscott. *The case against grammar correction in L2 writing classes.* Language learning, 46(2):327–369, 1996

John Truscott. *The case for "the case against grammar correction in L2 writing classes": A response to Ferris.* Journal of second language writing, 8(2):111–122, 1999

References

Feedback in CALL

- Trude Heift. *Corrective feedback and learner uptake in CALL*. ReCALL, 16(2):416–431, 2004

- Trude Heift. *Error-specific and individualised feedback in a Web-based language tutoring system: Do they read it?* ReCALL, 13(1):99–109, 2001

- Trude Heift, Phong Nguyen, and Volker Hegelheimer. *Technology-mediated corrective feedback*. The Cambridge handbook of corrective feedback in second language learning and teaching, 2021

- Trude Heift and Nina Vyatkina. *Technologies for teaching and learning L2 grammar*. The handbook of technology and second language teaching and learning, pages 26–44, 2017

Feedback Comment Generation

- Diana Galvan-Sosa, Steven Coyne, Keisuke Sakaguchi, and Kentaro Inui. *Towards grammatically-informed feedback comments*. 2023

References

- Kazuaki Hanawa, Ryo Nagata, and Kentaro Inui. *Exploring methods for generating feedback comments for writing learning*. In Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, pages 9719–9730, 2021

- Ryo Nagata, Masato Hagiwara, Kazuaki Hanawa, Masato Mita, Artem Chernodub, and Olena Nahorna. *Shared task on feedback comment generation for language learners*. In Proceedings of the 14th International Conference on Natural Language Generation, pages 320–324, 2021

- Ryo Nagata, Kentaro Inui, and Shin'ichiro Ishikawa. *Creating corpora for research in feedback comment generation*. In Proceedings of the Twelfth Language Resources and Evaluation Conference, pages 340–345, 2020

My publications relevant to this project

- 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, 2021

References

- Arianna Masciolini. *A query engine for L1-L2 parallel dependency treebanks*. In The 24rd Nordic Conference on Computational Linguistics, 2023