

Feedback for language learners with UD and GF

8th GF Summer School, 22.08.2023

Arianna Masciolini

The problem

In call applications, feedback is often:

- ❑ based on matching user input to an expected strings
- ❑ only available in English
- ❑ not helpful to understand what the mistake actually is

The goal

Developing an approach to generate feedback for grammatical (=morphosyntactical) errors.

The feedback should be:

- ❑ not dependent on a specific exercise format
- ❑ available for multiple L2s
- ❑ available in multiple L1s
- ❑ explicative and well suited to the learner level

L1-L2 Parallel Dependency Treebank as Learner Corpus

John Lee, Keying Li, Herman Leung

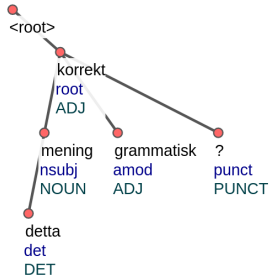
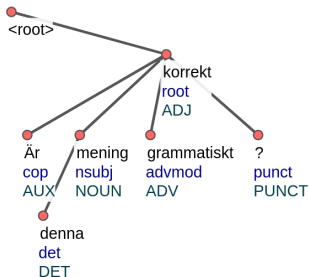
Department of Linguistics and Translation

City University of Hong Kong

jsylee@cityu.edu.hk, keyingli3-c@my.cityu.edu.hk, leung.hm@gmail.com

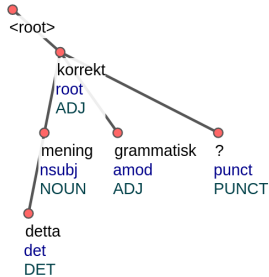
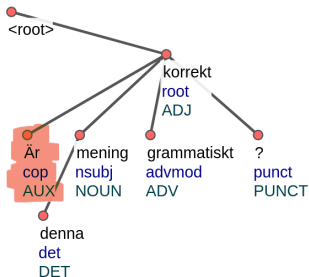
- ❖ L2 sentences // correction hypotheses
- ❖ no explicit error tagging, just **UD annotation**
 - ❖ better **interoperability** between learner corpora

L1-L2 treebanks: example



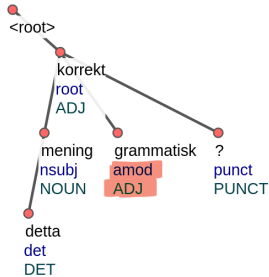
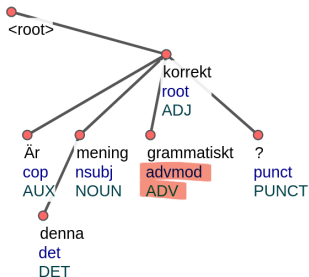
L1: "Är denna mening grammatiskt korrekt?" — L2: "detta mening korrekt grammatisk?"

L1-L2 treebanks: example



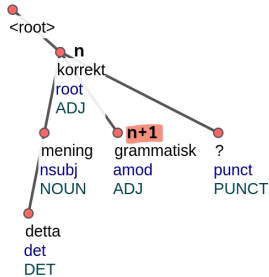
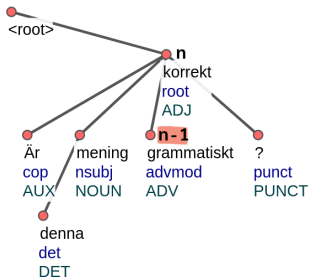
L1: "Är denna mening grammatiskt korrekt?" — L2: "detta mening korrekt grammatiskt?"

L1-L2 treebanks: example



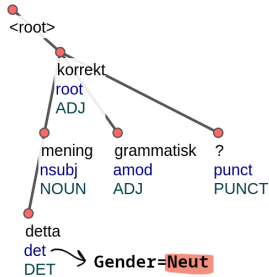
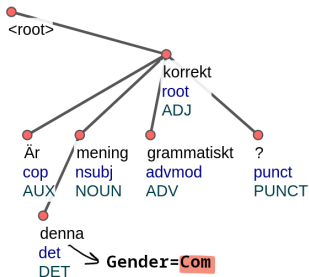
L1: "Är denna mening grammatiskt korrekt?" — L2: "detta mening korrekt grammatiskt?"

L1-L2 treebanks: example



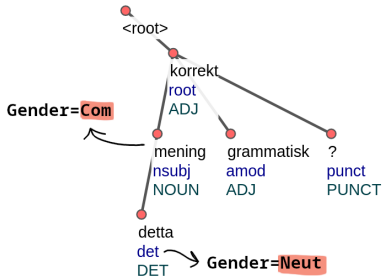
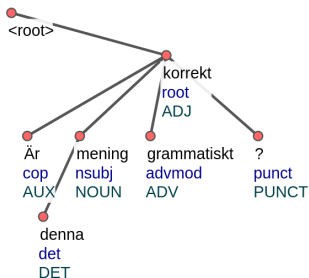
L1: "Är denna mening grammatiskt korrekt?" — L2: "detta mening korrekt grammatiskt?"

L1-L2 treebanks: example



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

L1-L2 treebanks: example



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

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
- ❑ back-and-forth MT to the learner's L1 can help

Back-and-forth translation

The screenshot shows a Google Translate interface with the following elements:

- Language selection: "Swedish (detected)" on the left and "English (UK)" on the right, with a double-headed arrow between them.
- Source text: "detta mening korrekt grammatisk?" with red wavy underlines under each word.
- Target text: "Is this sentence grammatically correct?"
- Section: "Alternatives:" followed by three options:
 - "Is this sentence correctly grammatical?"
 - "This sentence is correctly grammatical?"
 - "this sentence correctly grammatically?"
- Bottom icons: thumbs up/down, edit, copy, and share.
- Top right: "Glossary" button.

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?

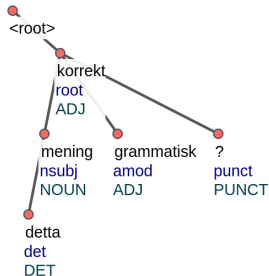
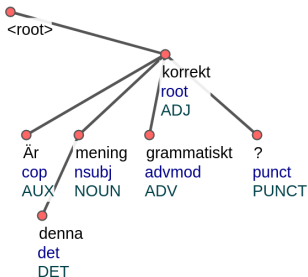
Steps

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 is more challenging.

Some (vague) ideas:

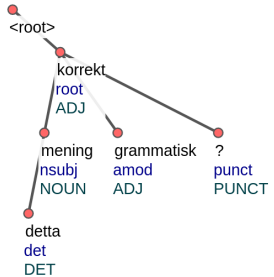
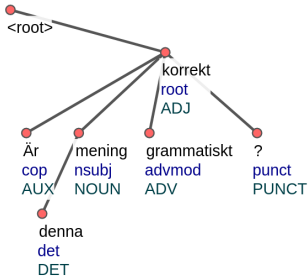
- ✚ just training a standard parser on a UD-annotated L2 corpus?
- ✚ L2 parsing “informed” by the L1 parse?

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

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 Machine Translation ?!

☆ Star ▾

machine-translation

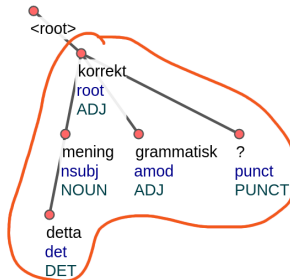
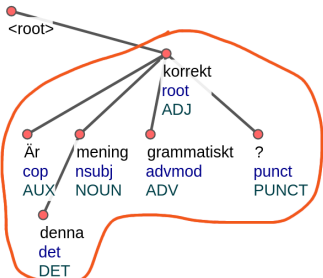
universal-dependencies

grammatical-framework

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

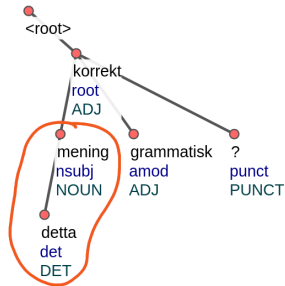
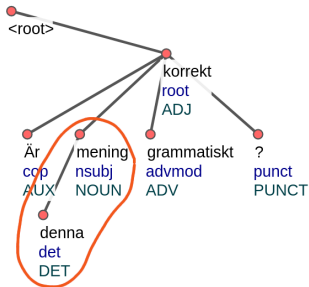
- ❏ finds word- and phrase-level correspondences in parallel UD treebanks
- ❏ designed to build translation lexica, but fairly configurable
- ❏ the L1-L2 case is arguably easier than the multilingual one

3.1 Error-correction pairs



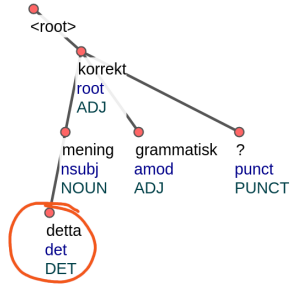
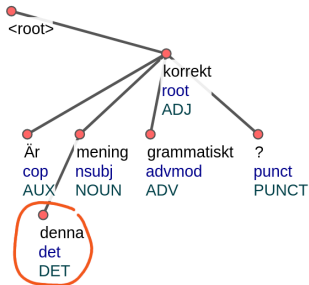
L1: "Är denna mening grammatiskt korrekt?" — L2: "detta mening korrekt grammatisk?"

3.1 Error-correction pairs



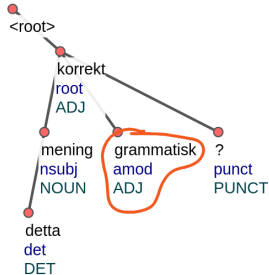
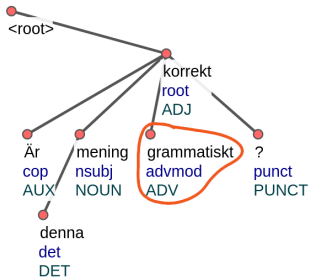
L1: "denna mening" — L2: "detta mening"

3.1 Error-correction pairs



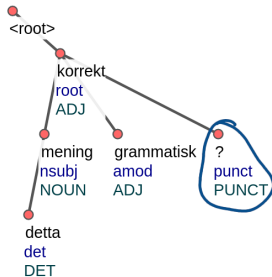
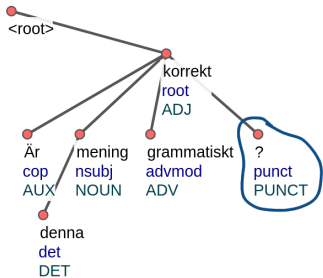
L1: "denna" — L2: "detta"

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**

3.1 Error-correction pairs

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

3.1 Error-correction pairs

- ❖ 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 (with 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

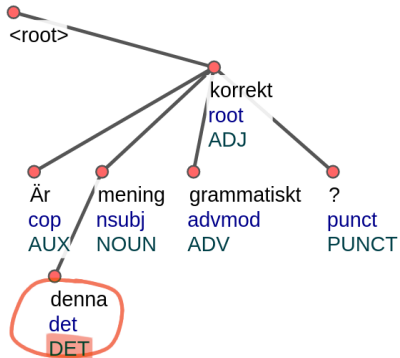
UD patterns in gf-ud

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

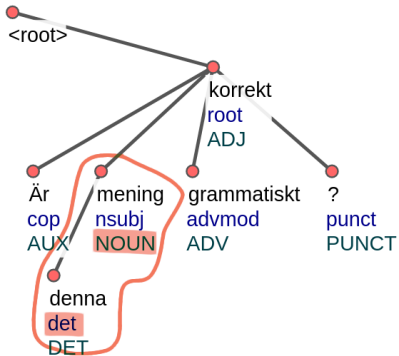


POS "DET"

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

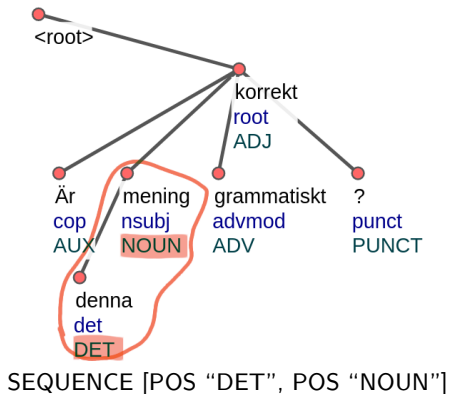


TREE (POS "NOUN") [DEPREL "det"]

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"]

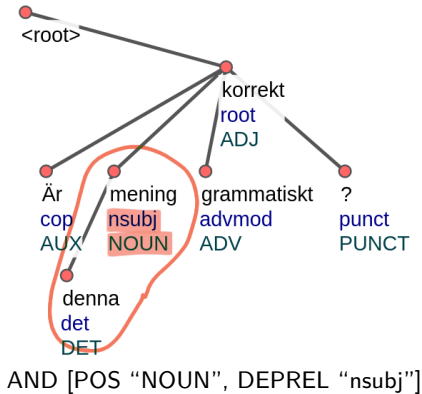
UD patterns in gf-ud



UD patterns in gf-ud

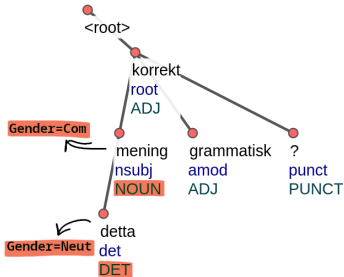
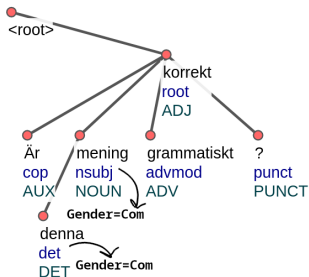
pattern type	example
single-token patterns	POS "DET"
tree patterns	TREE (POS "NOUN") [DEPREL "det"]
sequence patterns	SEQUENCE [POS "DET", POS "NOUN"]
logical operators	AND [POS "NOUN", DEPREL "nsubj"]

UD patterns in gf-ud



L1-L2 UD patterns

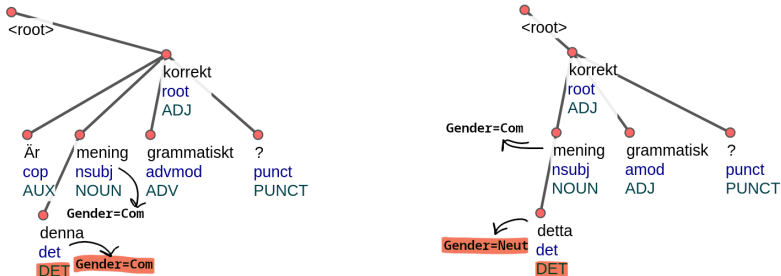
Many errors can be represented as UD patterns describing the L2



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

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

3.2 Error patterns

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

3.2 Error patterns

- ❖ 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

- ❖ 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!**

A more obvious thing to do with a query language

[← Go to NoDaLiDa 2023 Conference homepage](#)

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: Author: Visible To: Hidden From:

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 but already useful)

Other uses

Although this tool is meant for L1-L2 treebanks, it could be used with any parallel UD treebank!

Example

To find how Spanish future tenses are rendered in Finnish:

1. Align a Finnish-Spanish treebank
2. Search for alignments with future tenses (query: FEATS_ "Tense=Fut")
3. (extract morphosyntactic patterns from the results)

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

Feedback in CALL

“Är detta mening grammatiskt korrekt?”

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

... or any combination of the above!

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**

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"]]
```



*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"]])
```



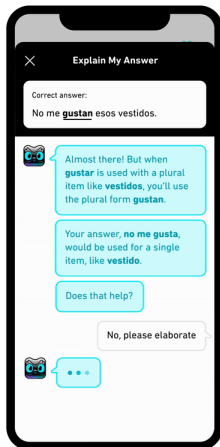
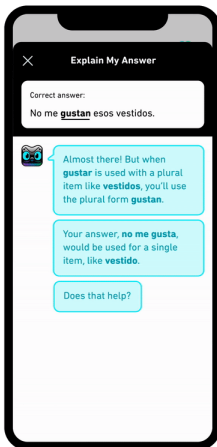
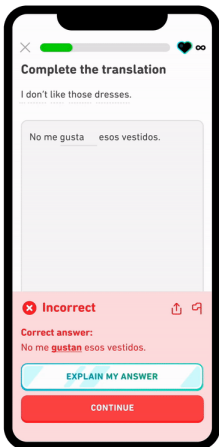
Pay attention to gender agreement!

Current status

step	status
1. obtain correction hypothesis	not my problem ;)
2. annotate learner sentence and correction in UD	up next, currently using a standard parser
3. extract error patterns	usable
4. generate feedback comments	still mere speculation

In recent news

Duolingo Explain My Answer



Thank you!