Using natural language processing in language documentation

Course: NLP for Endangered Languages of the Amazon. From a Uralic perspective. Lecture 5.

Jack Rueter, Niko Partanen,
Mika Hämäläinen & Khalid Alnajjar

Lecture 5

Two parts:

- Work on Oahpa as an example of community oriented language learning tool
- Accessing language documentation materials programmatically
    - Converting (last week)
    - Validation
    - Visualization
    - Research

Data planning for multiple use

- Address the needs of two communities
  - The language community and the researchers
- Dictionary databases
- Analyzers for linguists but also the language users

Analyzers for linguists but also the language users

- Prerequisites:
    - Morphology & Lexica
- Spell checking
    - Introducing language norms to a morphological description of the language
- Online dictionaries
    - With morphological analyses you can write any word form and find the article you are looking for. (No alphabetical order required)
- Intelligent Computer Assisted Language Learning (ICALL)
    - Here you can have the computer generate ONE normative form, but allow the students descriptive analyzers so they will get their choices accepted even if they are spelled poorly or “dialectal”

Oahpa = Learn!

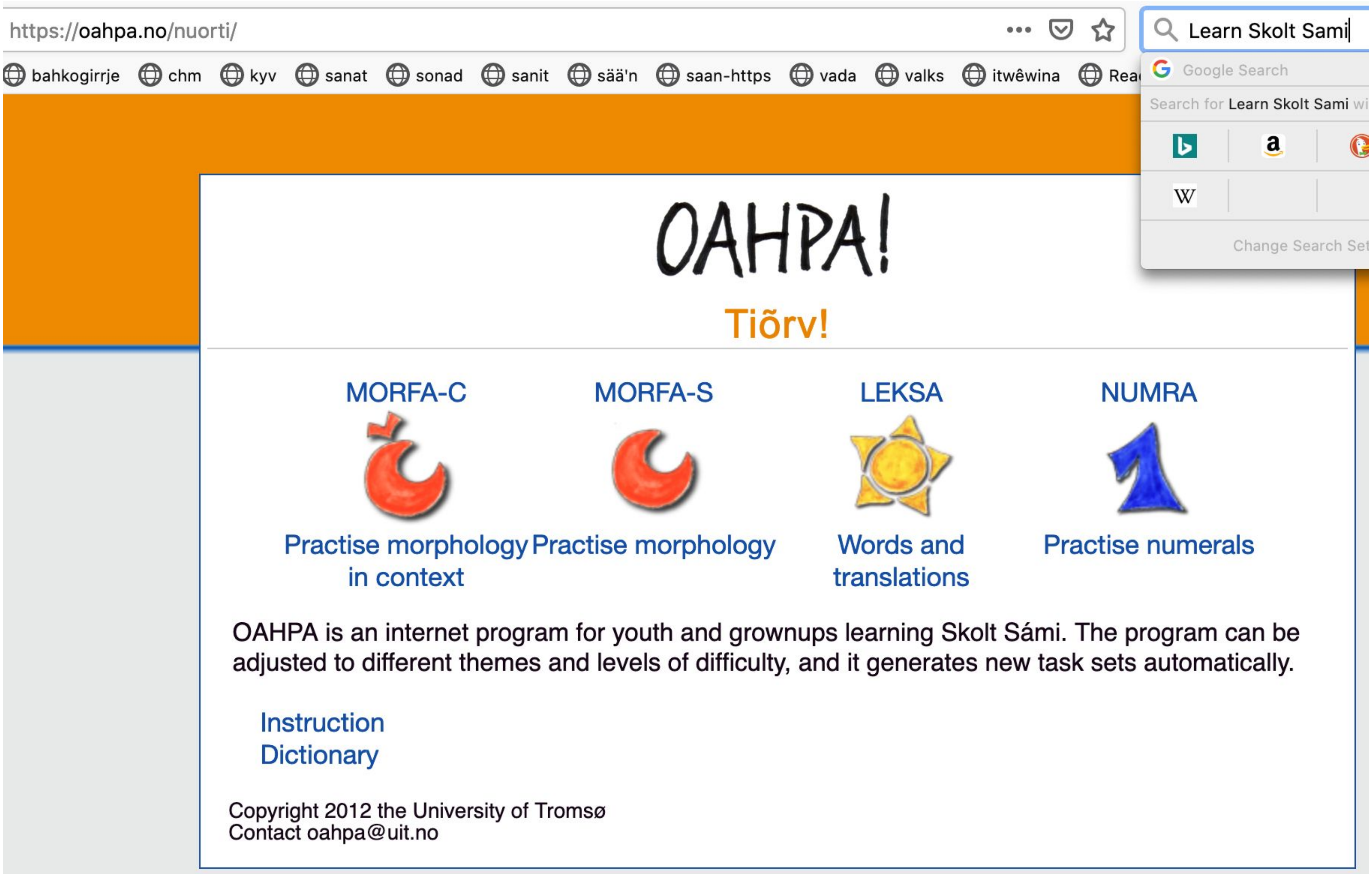
This is where Giellatekno realized they could not depend on English for a description of conjugation and declension of Sami-language verbs and nouns

Links to grammar articles, dictionaries

Use of published learning materials

Share data for the learning experience

- Numerals
- Lexicon
- Morphology
- Contextual morphology



OAHPA!

Tiõrv!

MORFA-C
Practise morphology in context

MORFA-S
Practise morphology

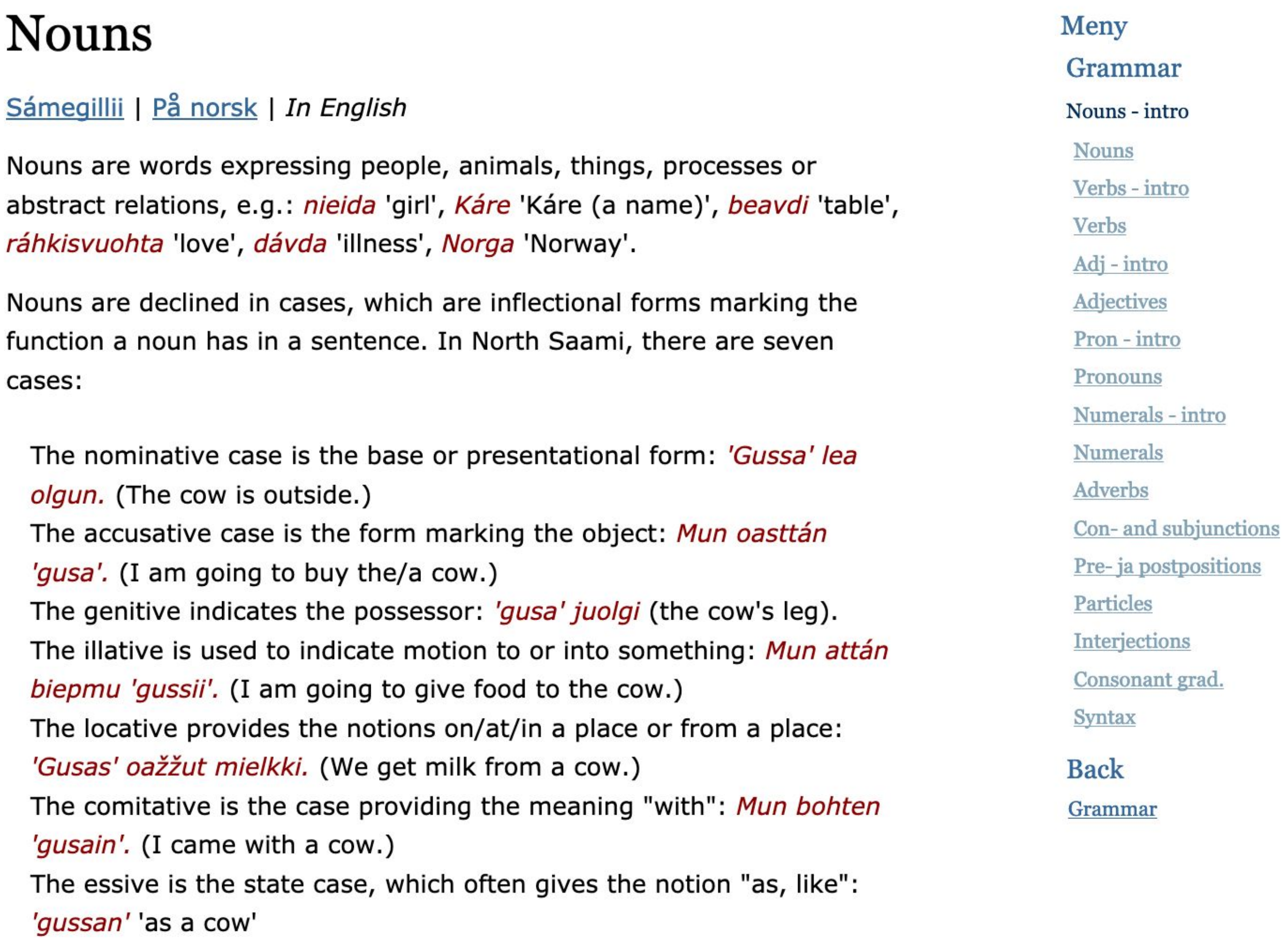
LEKSA
Words and translations

NUMRA
Practise numerals

OAHPA is an internet program for youth and grownups learning Skolt Sámi. The program can be adjusted to different themes and levels of difficulty, and it generates new task sets automatically.

Instruction
Dictionary

Copyright 2012 the University of Tromsø
Contact oahpa@uit.no



Nouns

Sámegillii | På norsk | *In English*

Nouns are words expressing people, animals, things, processes or abstract relations, e.g.: *nieida* 'girl', *Káre* 'Káre (a name)', *beavdi* 'table', *ráhkisvuohta* 'love', *dávda* 'illness', *Norga* 'Norway'.

Nouns are declined in cases, which are inflectional forms marking the function a noun has in a sentence. In North Saami, there are seven cases:

The nominative case is the base or presentational form: *'Gussa' lea olgun.* (The cow is outside.)
The accusative case is the form marking the object: *Mun oasttán 'gusa'.* (I am going to buy the/a cow.)
The genitive indicates the possessor: *'gusa' juolgi* (the cow's leg).
The illative is used to indicate motion to or into something: *Mun attán biepmu 'gussii'.* (I am going to give food to the cow.)
The locative provides the notions on/at/in a place or from a place: *'Gusas' oažžut mielkki.* (We get milk from a cow.)
The comitative is the case providing the meaning "with": *Mun bohten 'gusain'.* (I came with a cow.)
The essive is the state case, which often gives the notion "as, like": *'gussan'* 'as a cow'

Meny

Grammar

Nouns - intro

Nouns

Verbs - intro

Verbs

Adj - intro

Adjectives

Pron - intro

Pronouns

Numerals - intro

Numerals

Adverbs

Con- and subjunctions

Pre- ja postpositions

Particles

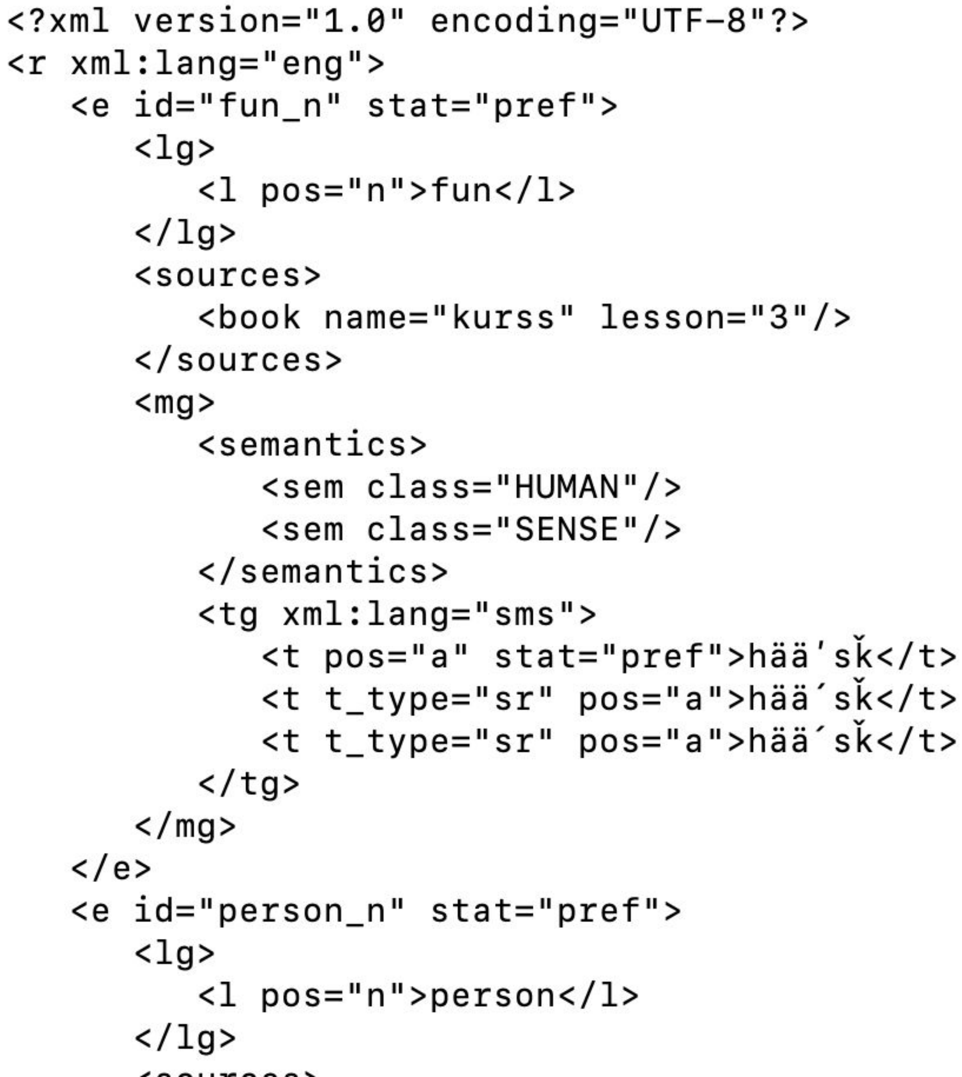
Interjections

Consonant grad.

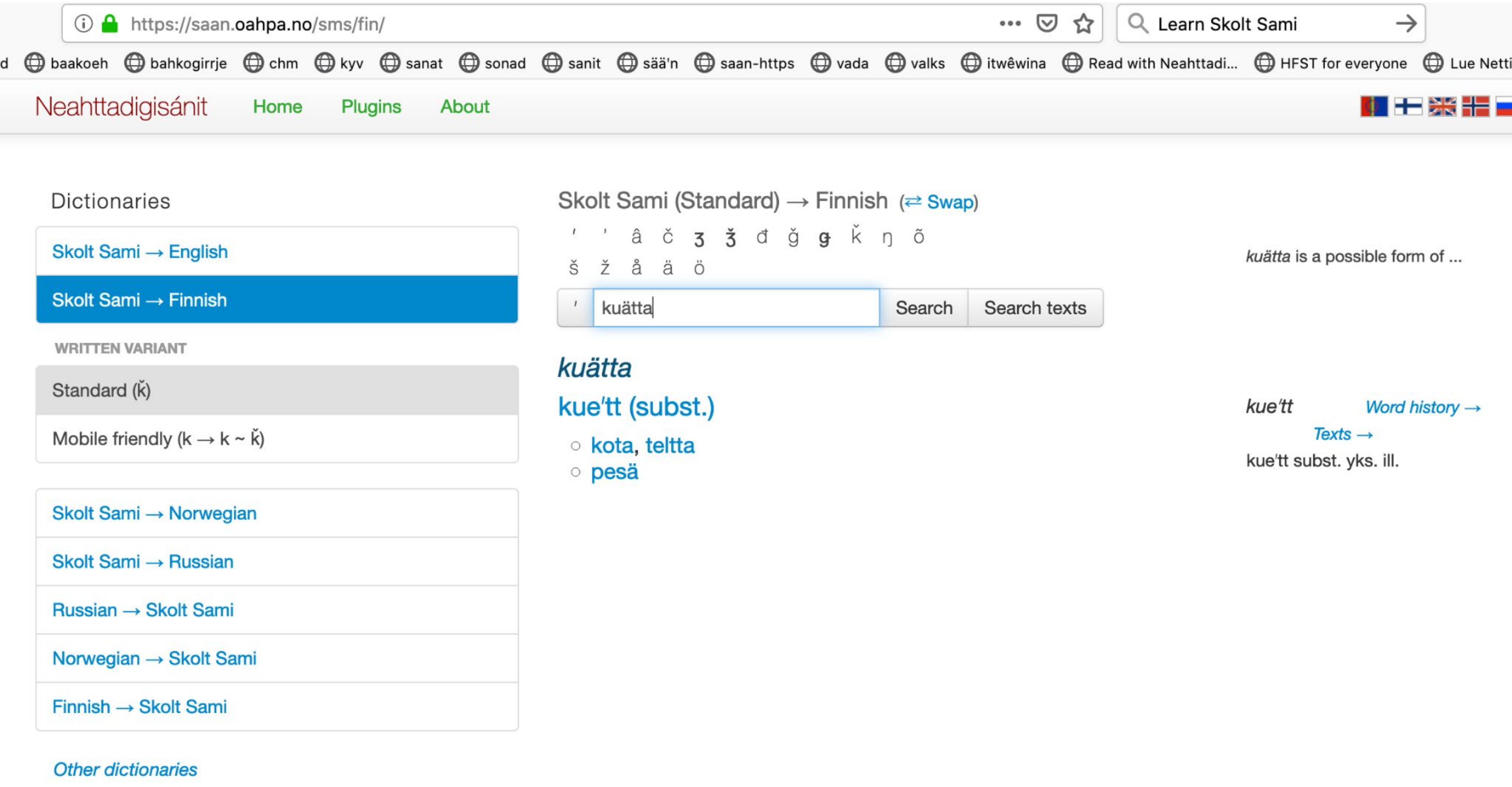
Syntax

Back

Grammar



```
<?xml version="1.0" encoding="UTF-8"?>
<r xml:lang="eng">
   <e id="fun_n" stat="pref">
      <lg>
         <l pos="n">fun</l>
      </lg>
      <sources>
         <book name="kurss" lesson="3"/>
      </sources>
      <mg>
         <semantics>
            <sem class="HUMAN"/>
            <sem class="SENSE"/>
         </semantics>
         <tg xml:lang="sms">
            <t pos="a" stat="pref">hääʹsǩ</t>
            <t t_type="sr" pos="a">hää´sǩ</t>
            <t t_type="sr" pos="a">hää´sǩ</t>
         </tg>
      </mg>
   </e>
   <e id="person_n" stat="pref">
      <lg>
         <l pos="n">person</l>
      </lg>
      <sources>
```



Dictionaries

Skolt Sami (Standard) → Finnish (⇄ Swap)

kuätta

kuätta

kue'tt (subst.)

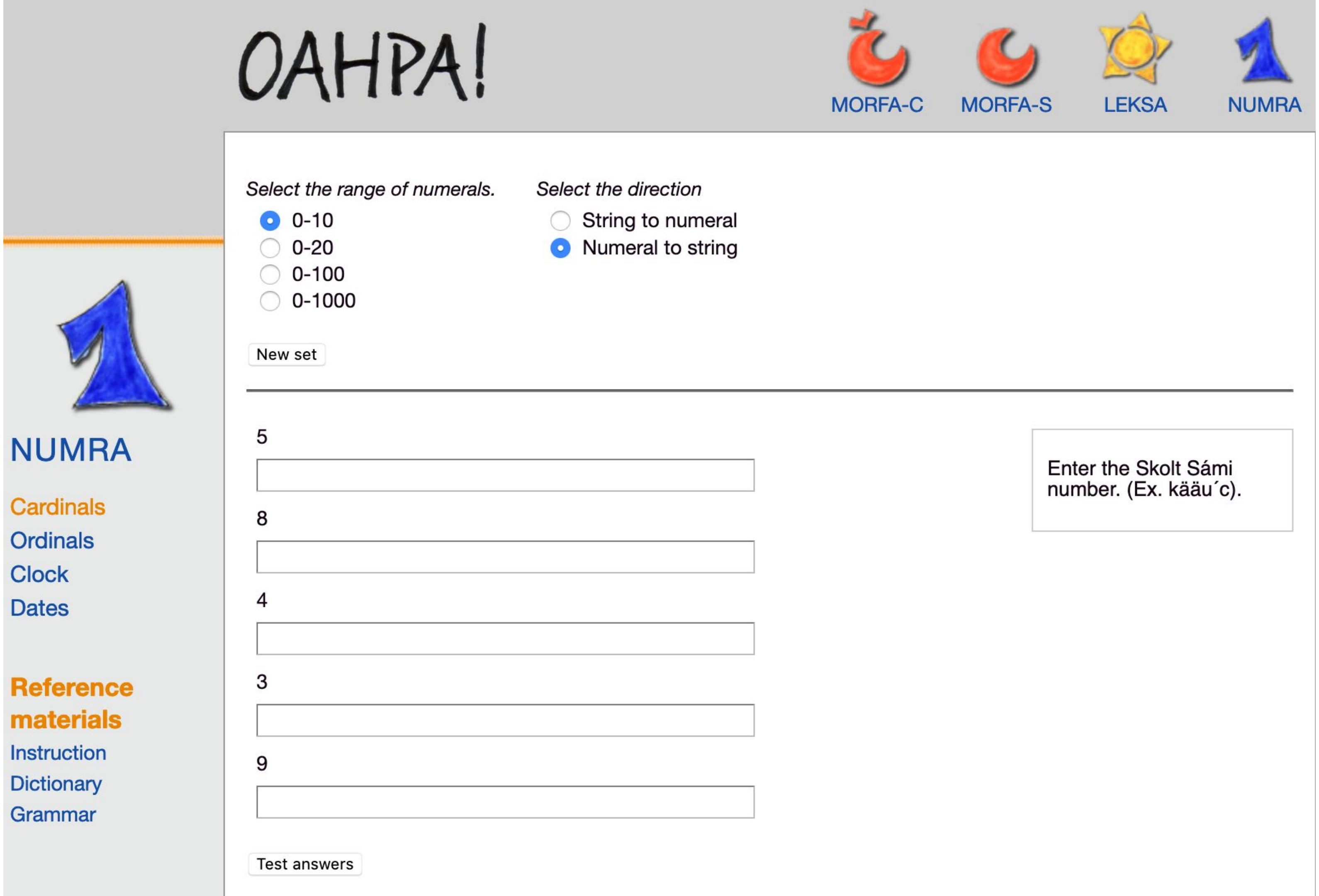
- kota, teltta
- pesä

kuätta is a possible form of ...

kue'tt *Word history →*

Texts →

kue'tt subst. yks. ill.



Select the range of numerals.

● 0-10
○ 0-20
○ 0-100
○ 0-1000

Select the direction

○ String to numeral
● Numeral to string

5

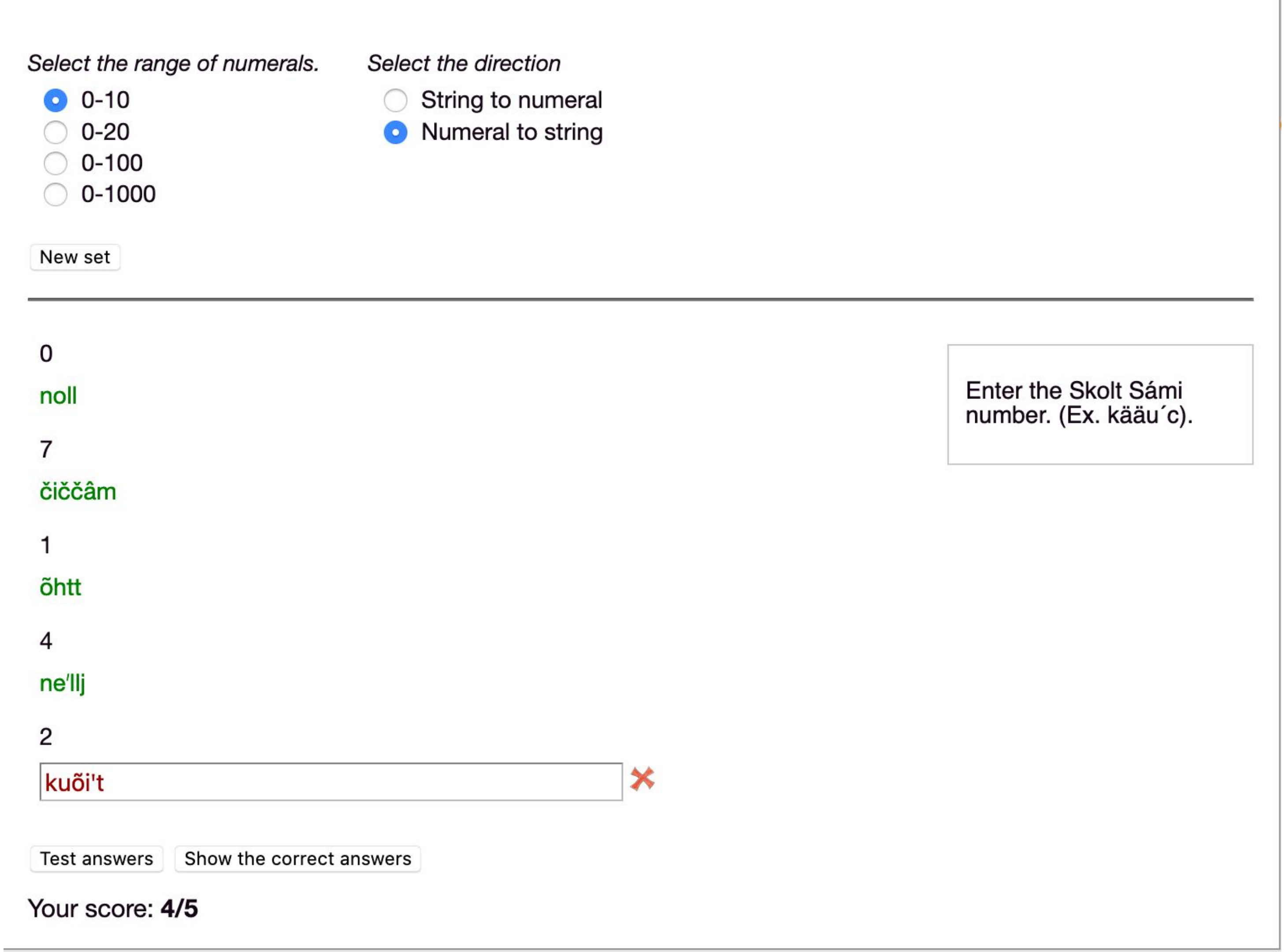
8

4

3

9

Enter the Skolt Sámi number. (Ex. kääu´c).



Select the range of numerals.

● 0-10
○ 0-20
○ 0-100
○ 0-1000

Select the direction

○ String to numeral
● Numeral to string

0

noll

7

čiččâm

1

õhtt

4

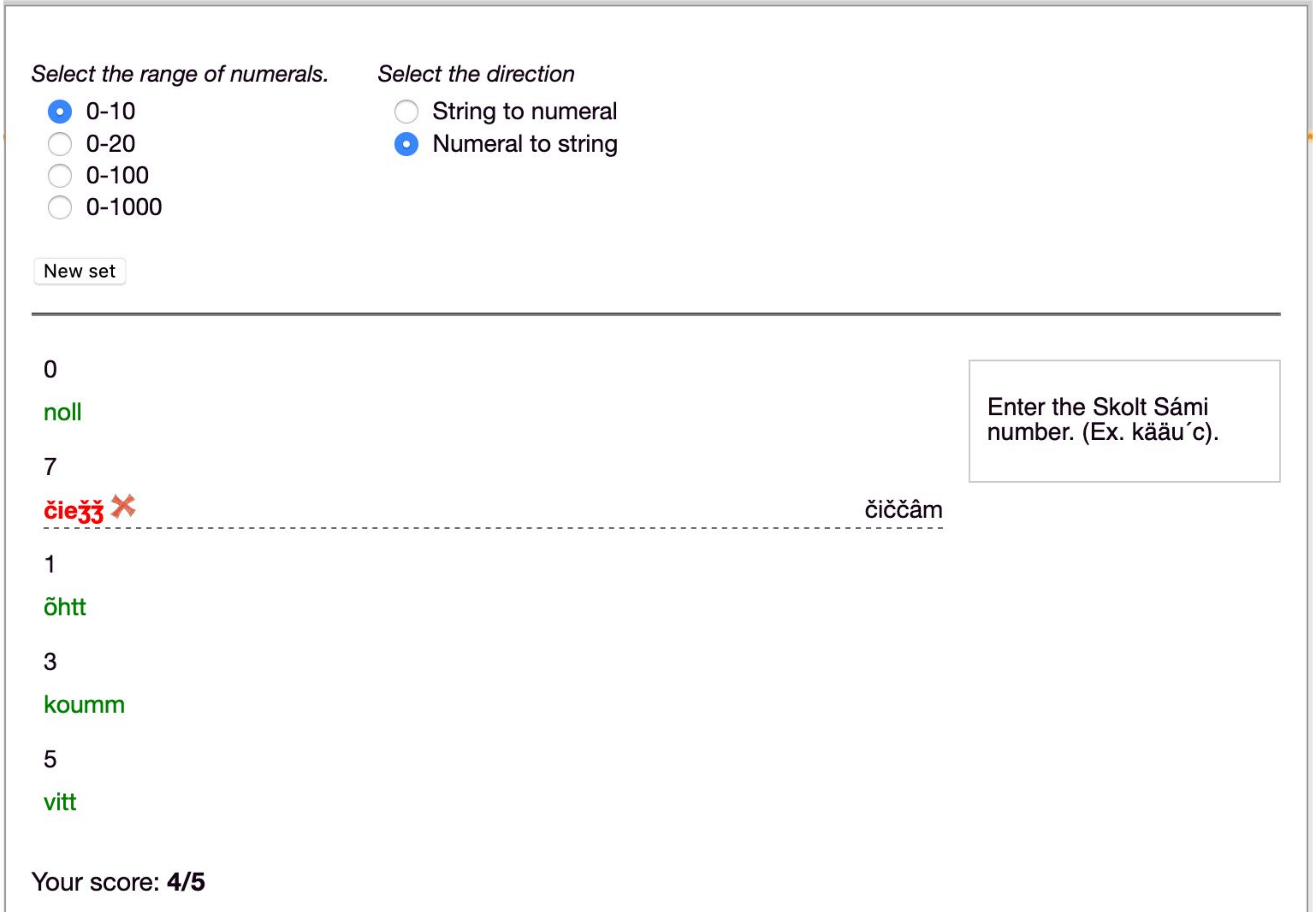
ne′llj

2

kuõi't ✗

Enter the Skolt Sámi number. (Ex. kääu´c).

Your score: **4/5**



Select the range of numerals.

● 0-10
○ 0-20
○ 0-100
○ 0-1000

Select the direction

○ String to numeral
● Numeral to string

0

noll

7

čieǯǯ ✗ čiččâm

1

õhtt

3

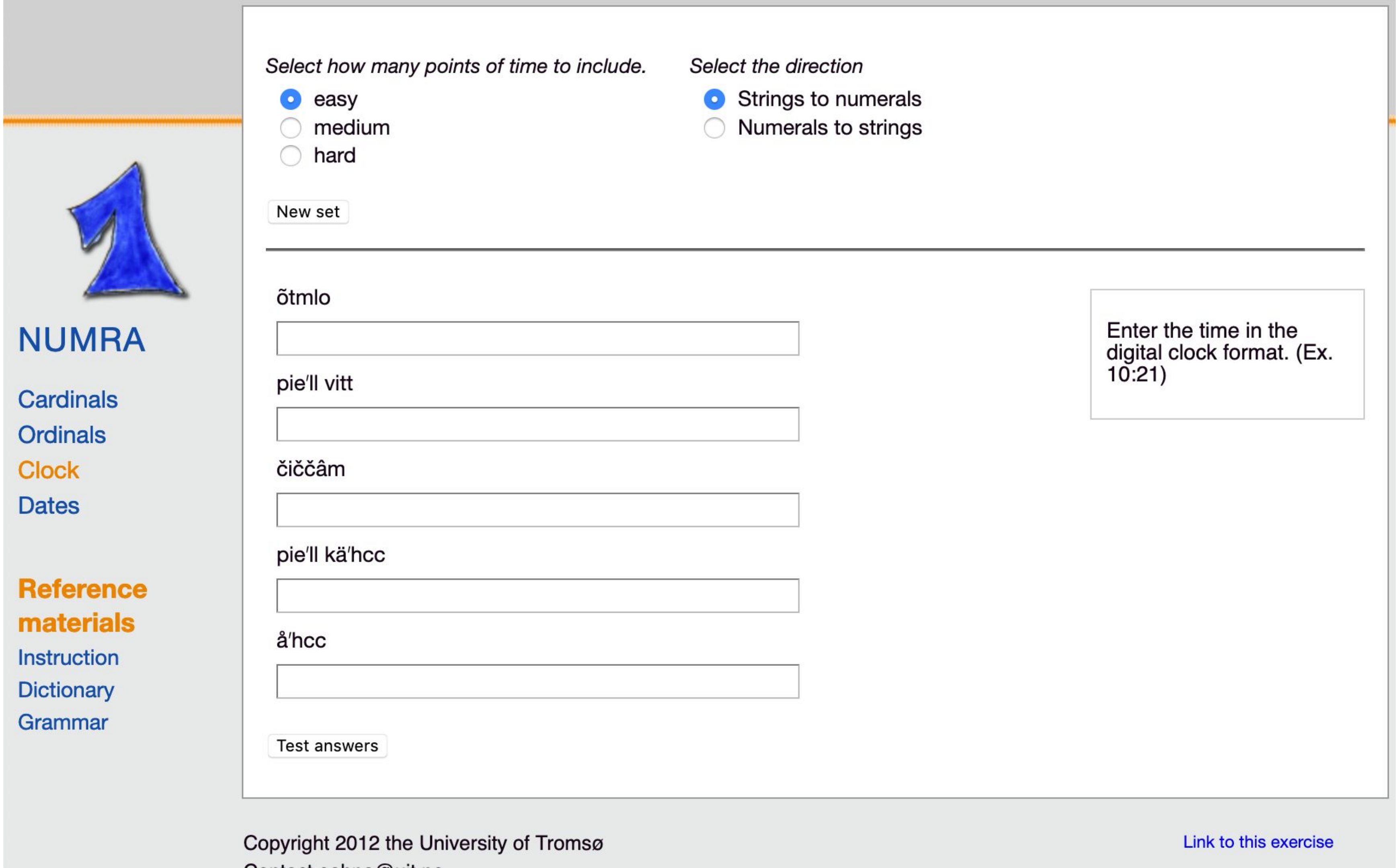
koumm

5

vitt

Enter the Skolt Sámi number. (Ex. kääu´c).

Your score: **4/5**



Select how many points of time to include.

● easy
○ medium
○ hard

Select the direction

● Strings to numerals
○ Numerals to strings

õtmlo

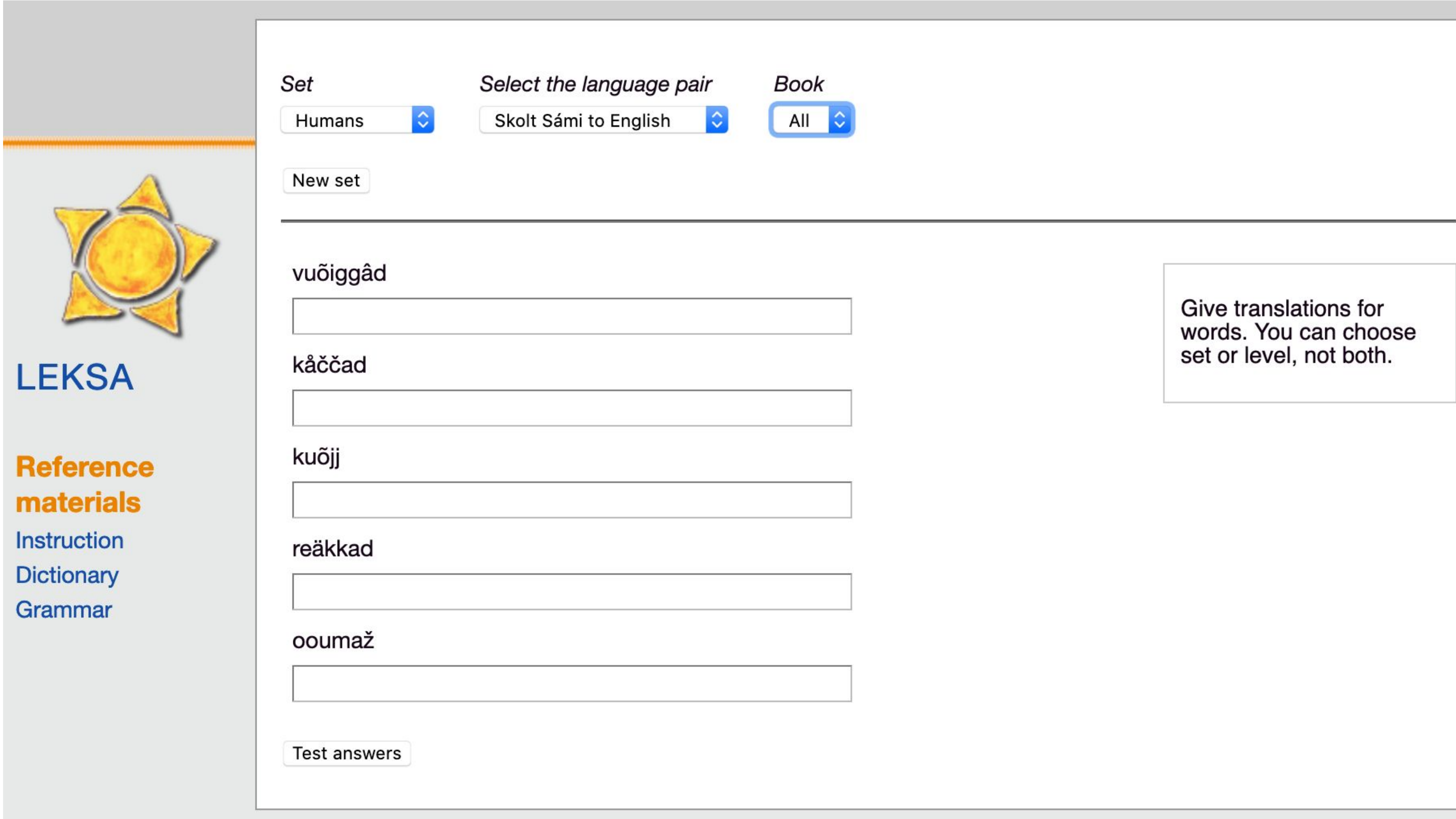
pie′ll vitt

čiččâm

pie′ll kä′hcc

å′hcc

Enter the time in the digital clock format. (Ex. 10:21)



Set: Humans

Select the language pair: Skolt Sámi to English

Book: All

vuõiggâd

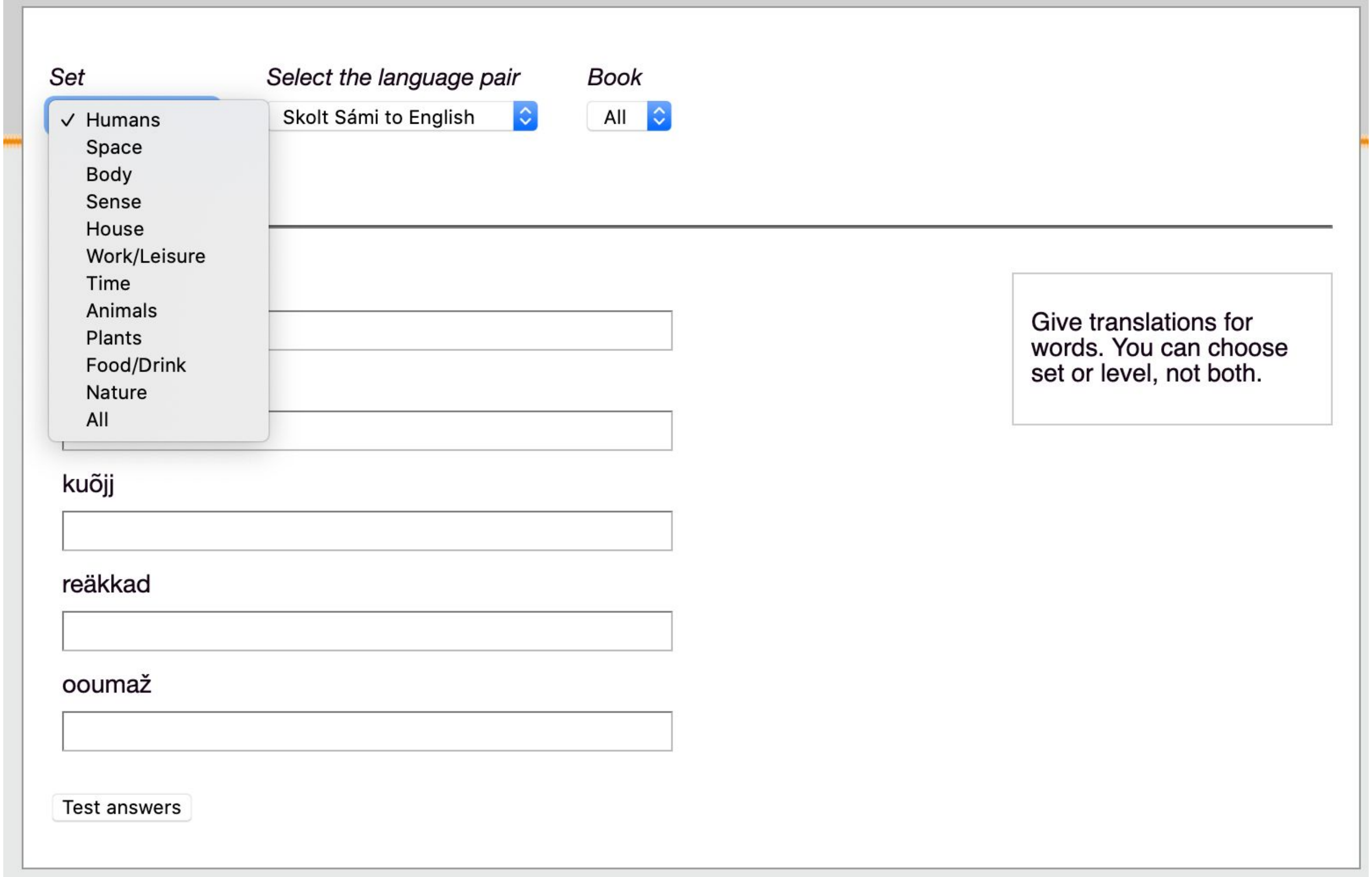
kåččad

kuõjj

reäkkad

oumaž

Give translations for words. You can choose set or level, not both.



Set ✓ Humans

Select the language pair Skolt Sámi to English

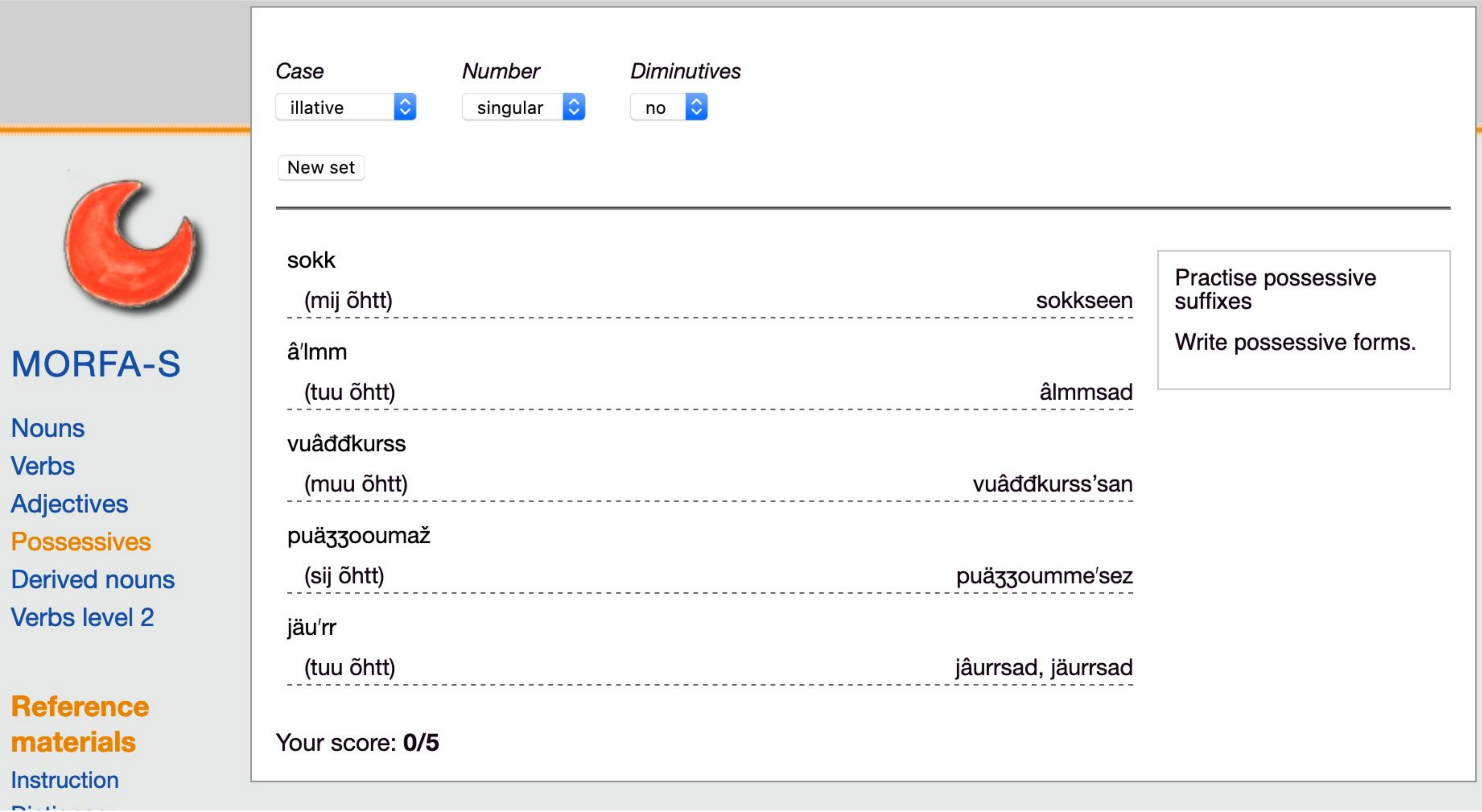
Book All

kuõjj

reäkkad

ooumaž

Give translations for words. You can choose set or level, not both.



Case: illative

Number: singular

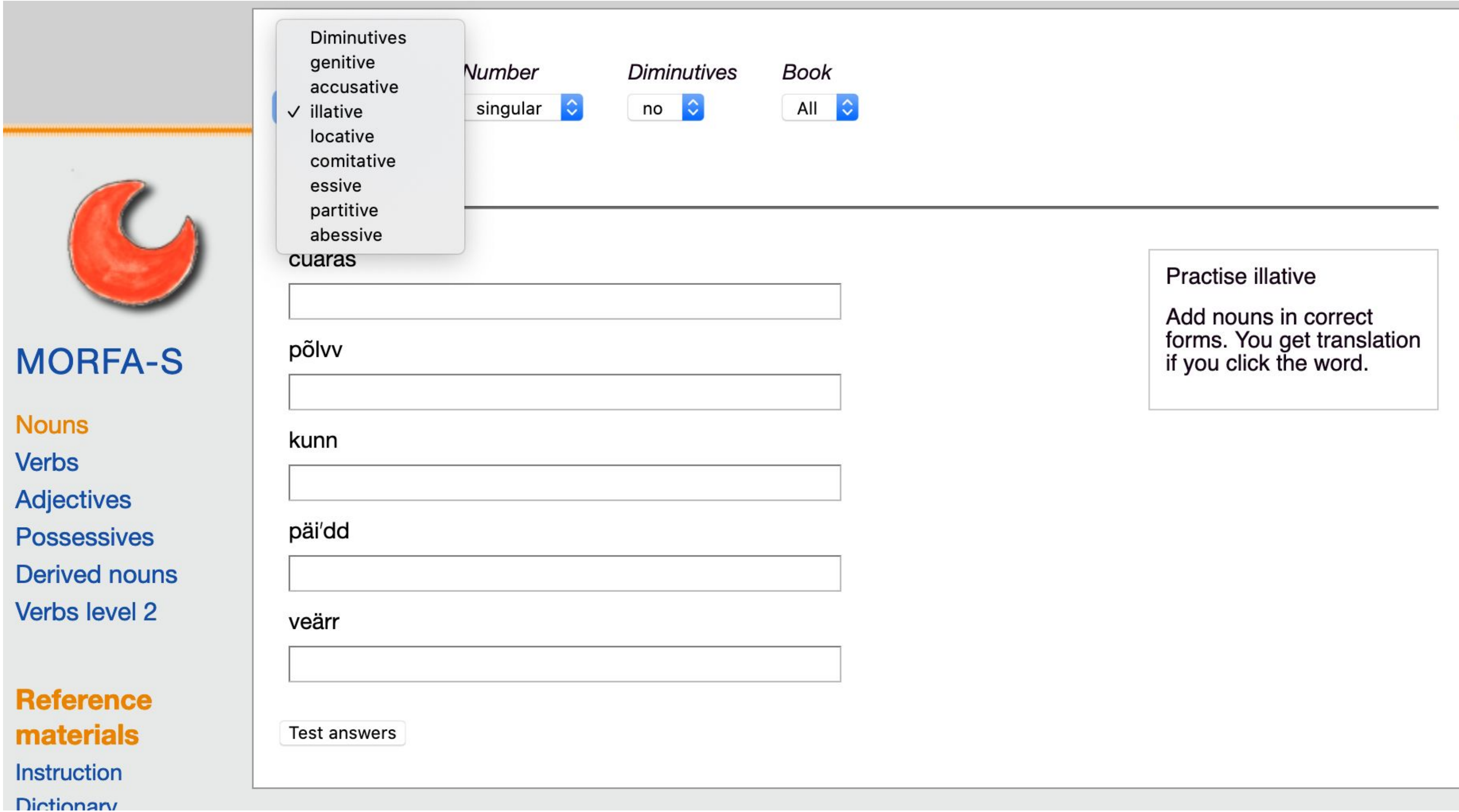
Diminutives: no

sokk	
(mij õhtt)	sokkseen
âʹlmm	
(tuu õhtt)	âlmmsad
vuâđđkurss	
(muu õhtt)	vuâđđkurss’san
puäʒʒooumaž	
(sij õhtt)	puäʒʒoummeʹsez
jäuʹrr	
(tuu õhtt)	jâurrsad, jäurrsad

Your score: **0/5**

Practise possessive suffixes

Write possessive forms.



Number singular

Diminutives no

Book All

cuaras

põlvv

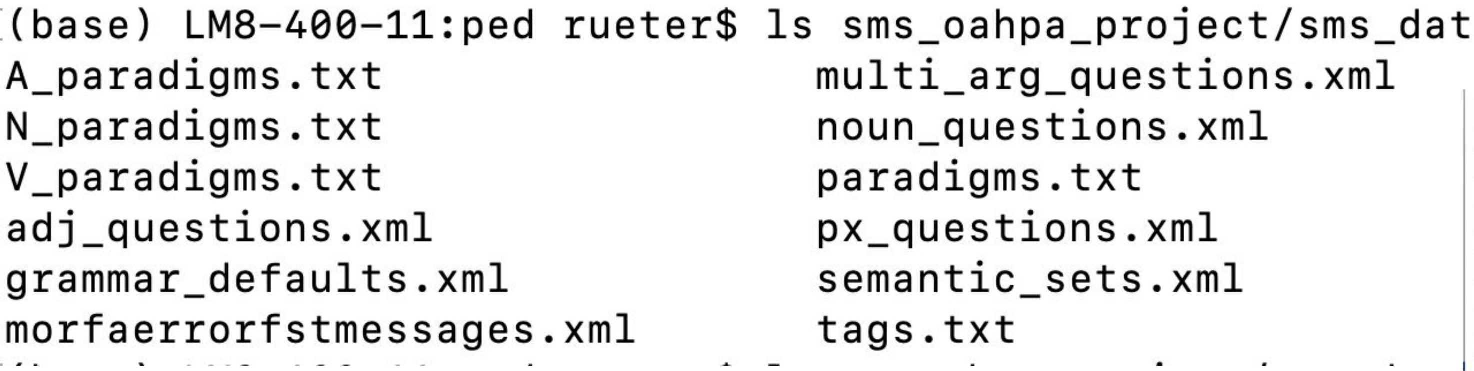
kunn

päi'dd

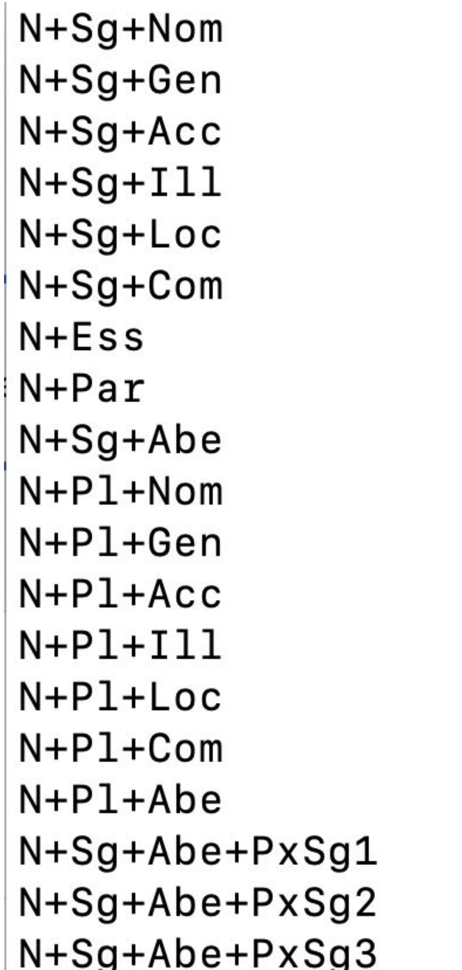
veärr

Practise illative

Add nouns in correct forms. You get translation if you click the word.



```
(base) LM8-400-11:ped rueter$ ls sms_oahpa_project/sms_data/meta_data/
A_paradigms.txt                 multi_arg_questions.xml
N_paradigms.txt                 noun_questions.xml
V_paradigms.txt                 paradigms.txt
adj_questions.xml               px_questions.xml
grammar_defaults.xml            semantic_sets.xml
morfaerrorfstmessages.xml       tags.txt
```



```
N+Sg+Nom
N+Sg+Gen
N+Sg+Acc
N+Sg+Ill
N+Sg+Loc
N+Sg+Com
N+Ess
N+Par
N+Sg+Abe
N+Pl+Nom
N+Pl+Gen
N+Pl+Acc
N+Pl+Ill
N+Pl+Loc
N+Pl+Com
N+Pl+Abe
N+Sg+Abe+PxSg1
N+Sg+Abe+PxSg2
N+Sg+Abe+PxSg3
```

Thoughts for utilization of resources

- Provide a student-project grammar, where individuals can contribute.
  - This could serve for study points, and be offered to the early learners
- Use the vocabularies from your textbooks for a list of words to
  - Translate & Inflect
  - A list of all words used in texts
    - This will also help locate lesser researched word forms
    - Study materials can hopefully be digital and they might be used for improving tools (analyzers, spellcheckers, translations)
    - Language learners are an important part of the community
- Make the infrastructure available in the native language, too.

ELAN corpora

- To be able to edit the transcription next to the audio and video is necessary
- ELAN is a very good tool for this
- There are other alternatives, and if something works well, that’s good too!

- ELAN’s flexibility is one of it’s curses
- Tier structures can be indefinitely flexible
- For ELAN, files that deviate from project’s structure are alright
- For researcher use this is often a problem

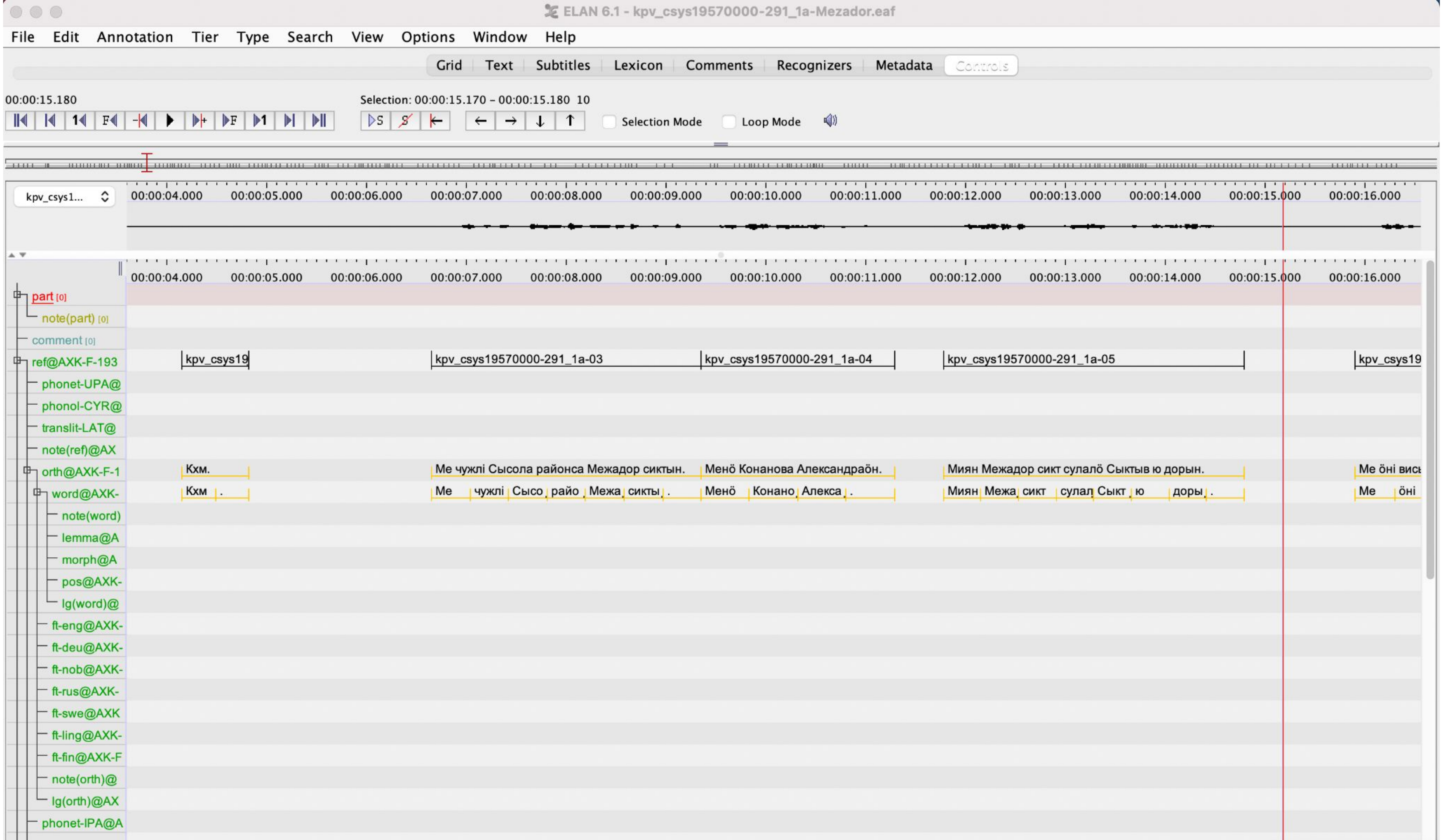
Usual scenario: We decide to change the project template

Usually results in some files being in the old template, the others in new

Can be less cumbersome than this – no context is the same

Most of the largest messes I have been in have resulted from manual editing – but maybe others are less lousy with their files!

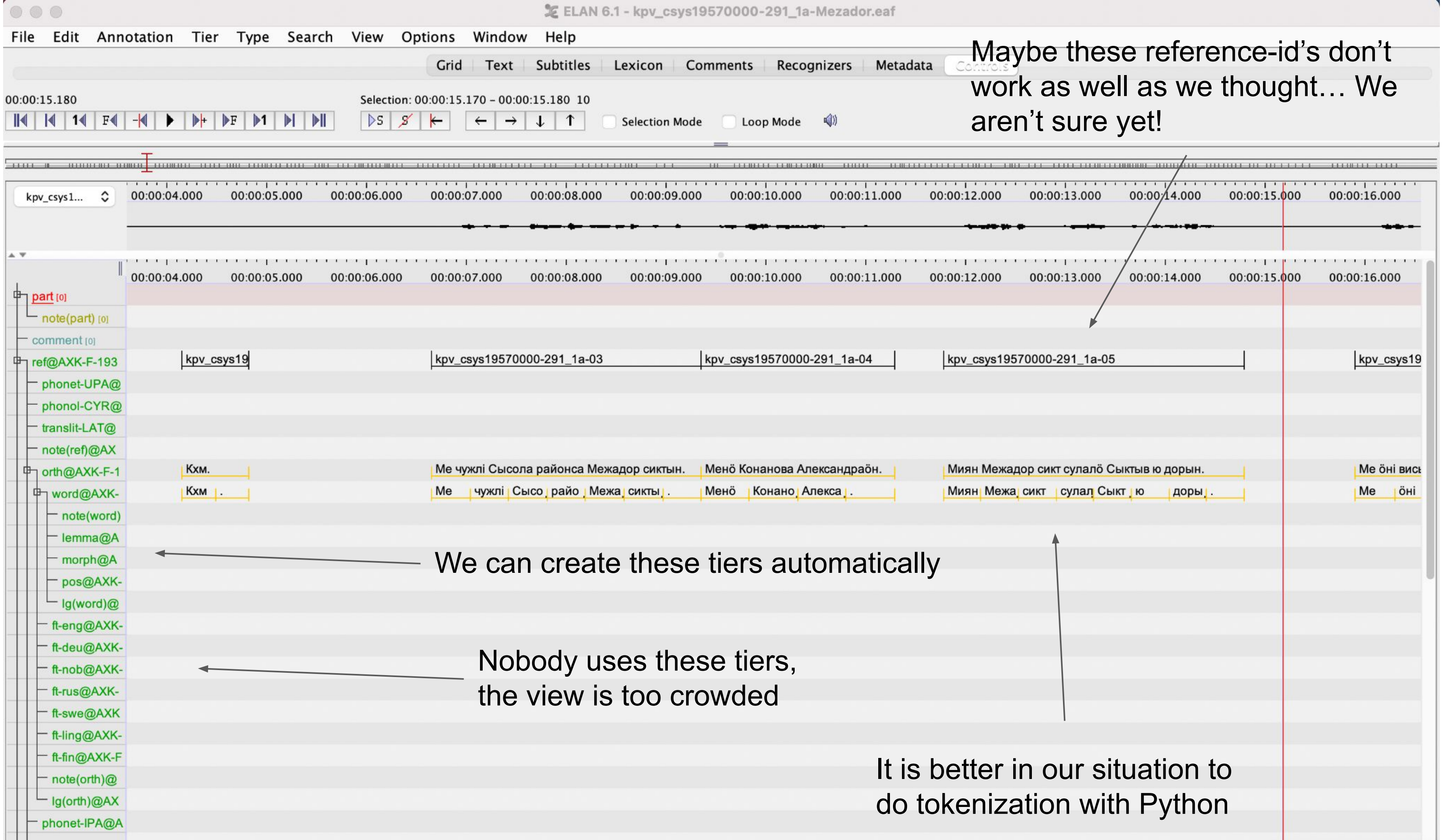
I took a small example from the Komi files I created in 2014, these are recordings from 1960s and 1970s – stored in the course repository in *corpus* folder



ref@AXK-F-193: kpv_csys19 | kpv_csys19570000-291_1a-03 | kpv_csys19570000-291_1a-04 | kpv_csys19570000-291_1a-05 | kpv_csys19

orth@AXK-F-1: Кхм. | Ме чужлі Сысола районса Межадор сиктын. | Менӧ Конанова Александраӧн. | Миян Межадор сикт сулалӧ Сыктыв ю дорын. | Ме ӧні вись

word@AXK-: Кхм | . | Ме | чужлі | Сысо | райо | Межа | сикты | . | Менӧ | Конано | Алекса | . | Миян | Межа | сикт | сулал | Сыкт | ю | доры | . | Ме | ӧні



kpv_csys19 | kpv_csys19570000-291_1a-03 | kpv_csys19570000-291_1a-04 | kpv_csys19570000-291_1a-05 | kpv_csys19

Кхм. | Ме чужлі Сысола районса Межадор сиктын. | Менӧ Конанова Александраӧн. | Миян Межадор сикт суланӧ Сыктыв ю дорын. | Ме ӧні вись

Кхм | . | Ме | чужлі | Сысо | райо | Межа | сикты | . | Менӧ | Конано | Алекса | . | Миян | Межа | сикт | сулал | Сыкт | ю | доры | . | Ме | ӧні

Maybe these reference-id's don't work as well as we thought… We aren't sure yet!

We can create these tiers automatically

Nobody uses these tiers, the view is too crowded

It is better in our situation to do tokenization with Python

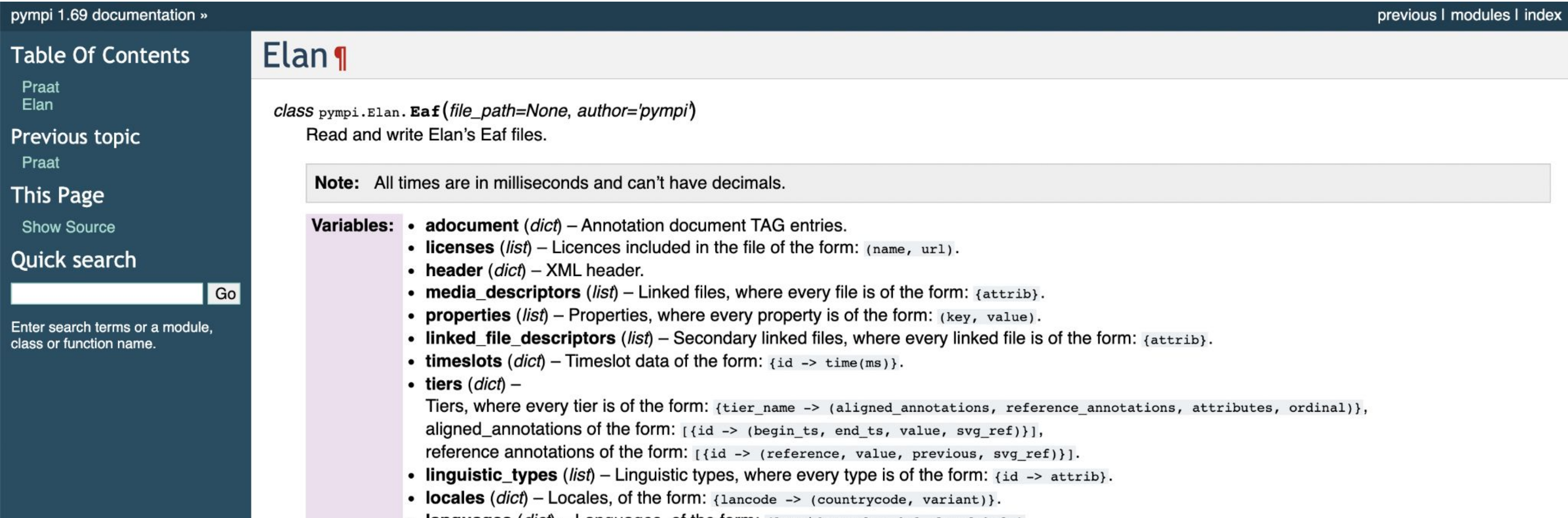
Possibilities

We remove the tiers manually, or we write a script that removes the tiers

Hiding the tiers works too, but that depends from pfsx files

Pympi is a very good alternative

https://dopefishh.github.io/pympi/Elan.html


pympi 1.69 documentation »
previous | modules | index
Table Of Contents
Praat
Elan
Previous topic
Praat
This Page
Show Source
Quick search
Go
Enter search terms or a module, class or function name.

Elan¶

class `pympi.Elan.`**`Eaf`**(*file_path=None*, *author='pympi'*)

Read and write Elan's Eaf files.

Note: All times are in milliseconds and can't have decimals.

Variables:

- **adocument** (*dict*) – Annotation document TAG entries.
- **licenses** (*list*) – Licences included in the file of the form: `(name, url)`.
- **header** (*dict*) – XML header.
- **media_descriptors** (*list*) – Linked files, where every file is of the form: `{attrib}`.
- **properties** (*list*) – Properties, where every property is of the form: `(key, value)`.
- **linked_file_descriptors** (*list*) – Secondary linked files, where every linked file is of the form: `{attrib}`.
- **timeslots** (*dict*) – Timeslot data of the form: `{id -> time(ms)}`.
- **tiers** (*dict*) –
  Tiers, where every tier is of the form: `{tier_name -> (aligned_annotations, reference_annotations, attributes, ordinal)}`,
  aligned_annotations of the form: `[{id -> (begin_ts, end_ts, value, svg_ref)}]`,
  reference annotations of the form: `[{id -> (reference, value, previous, svg_ref)}]`.
- **linguistic_types** (*list*) – Linguistic types, where every type is of the form: `{id -> attrib}`.
- **locales** (*dict*) – Locales, of the form: `{lancode -> (countrycode, variant)}`.

Installation

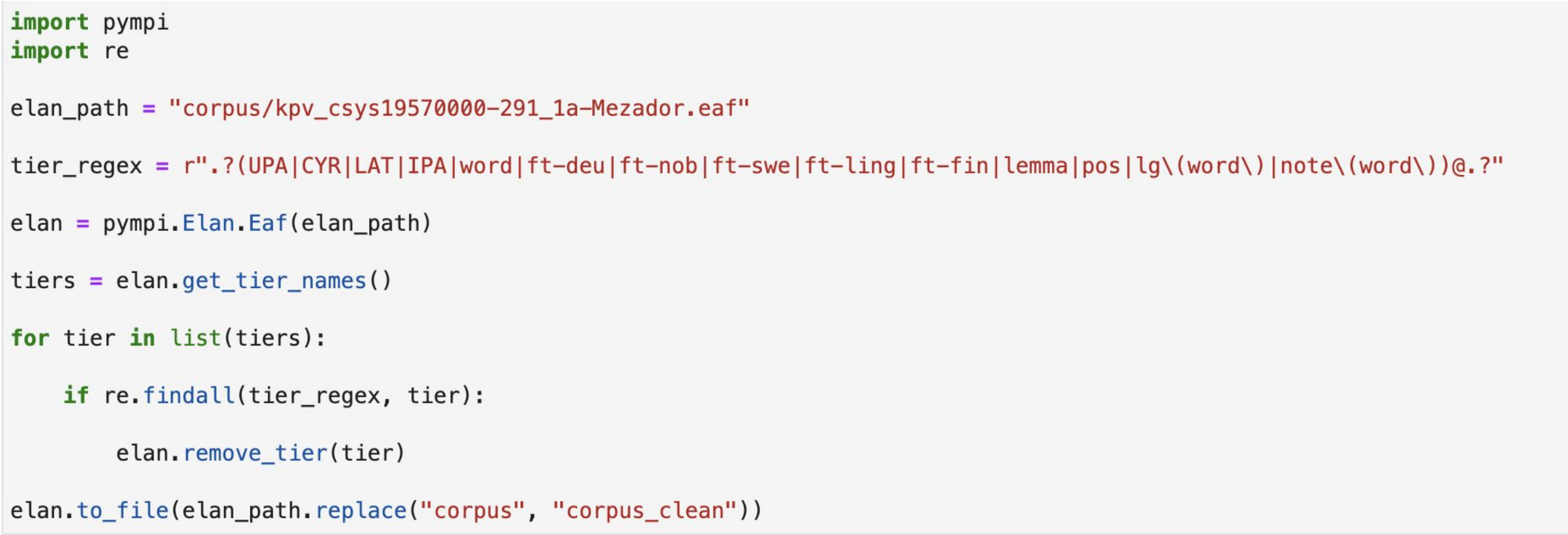
```
pip install pympi-ling
```

Basic use

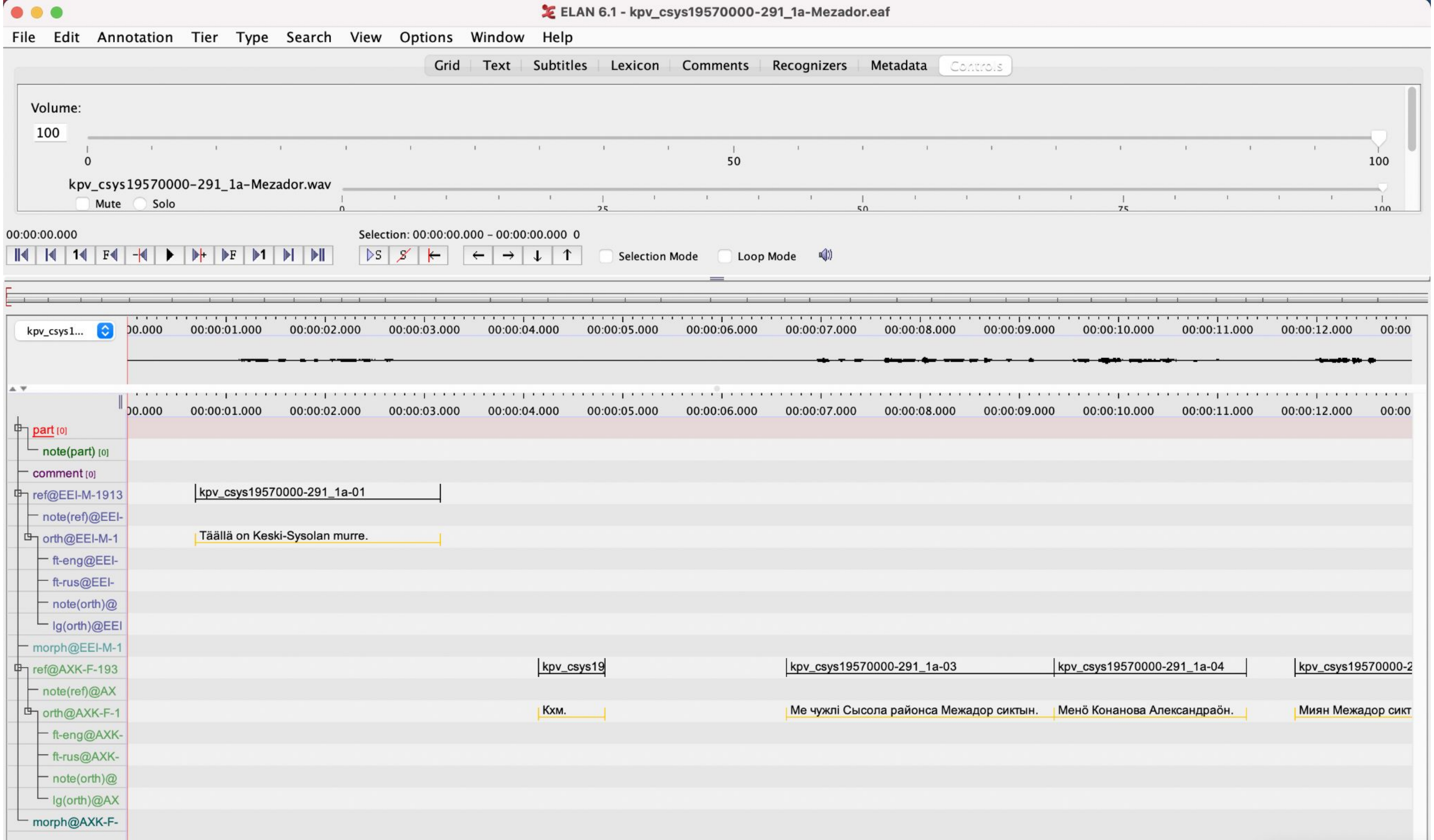
```
import pympi
elan = pympi.Elan.Eaf("corpus/elan_file.eaf")
elan.get_tier_names(…)
elan.rename_tier(…)
elan.remove_tier(…)
elan.to_file("corpus/edited_elan_file.eaf")
```

More complex example: We find all those useless tiers by their name in regex
And then we remove them, and save the new file into a new directory

We can also overwrite the file, if we know everything is ok :)



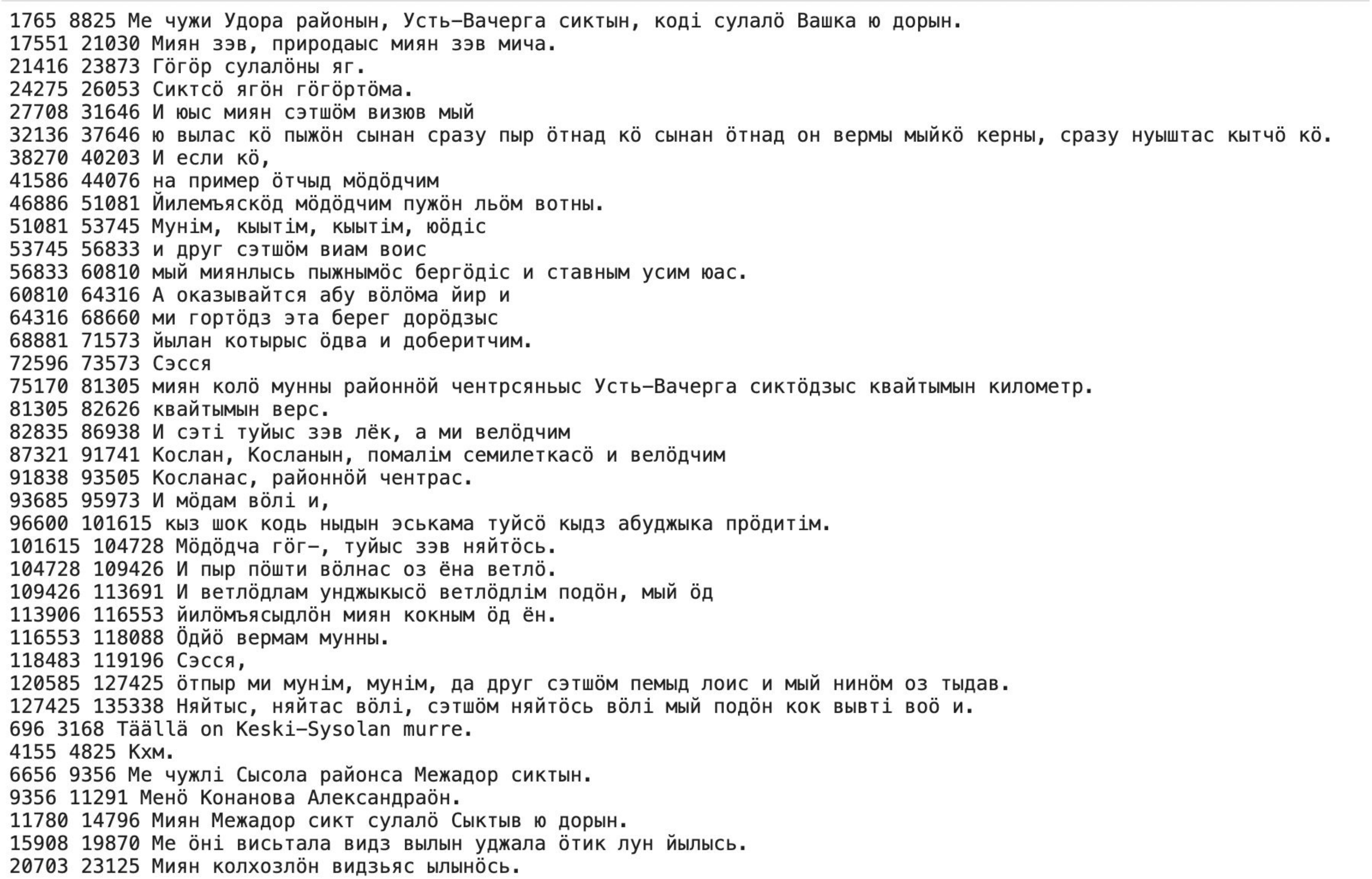
```
import pympi
import re

elan_path = "corpus/kpv_csys19570000-291_1a-Mezador.eaf"

tier_regex = r".?(UPA|CYR|LAT|IPA|word|ft-deu|ft-nob|ft-swe|ft-ling|ft-fin|lemma|pos|lg\(word\)|note\(word\))@.?"

elan = pympi.Elan.Eaf(elan_path)

tiers = elan.get_tier_names()

for tier in list(tiers):

    if re.findall(tier_regex, tier):

        elan.remove_tier(tier)

elan.to_file(elan_path.replace("corpus", "corpus_clean"))
```



part [0]
note(part) [0]
comment [0]
ref@EEI-M-1913 — kpv_csys19570000-291_1a-01
note(ref)@EEI-
orth@EEI-M-1 — Täällä on Keski-Sysolan murre.
ft-eng@EEI-
ft-rus@EEI-
note(orth)@
lg(orth)@EEI
morph@EEI-M-1
ref@AXK-F-193 — kpv_csys19 | kpv_csys19570000-291_1a-03 | kpv_csys19570000-291_1a-04 | kpv_csys19570000-2
note(ref)@AX
orth@AXK-F-1 — Кхм. | Ме чужлі Сысола районса Межадор сиктын. | Менӧ Конанова Александраӧн. | Миян Межадор сикт
ft-eng@AXK-
ft-rus@AXK-
note(orth)@
lg(orth)@AX
morph@AXK-F-

In our convention the orthographic transcription is on tier type **orthT**

```
elan = pympi.Elan.Eaf(file)
tiers = elan.get_tier_ids_for_linguistic_type("orthT")
for tier in tiers:
    annotations = elan.get_annotation_data_for_tier(tier)
    for annotation in annotations:
        print(annotation)
```



```
1765 8825 Ме чужи Удора районын, Усть-Вачерга сиктын, коді сулалö Вашка ю дорын.
17551 21030 Миян зэв, природаыс миян зэв мича.
21416 23873 Гöгöр сулалöны яг.
24275 26053 Сиктсö ягöн гöгöртöма.
27708 31646 И юыс миян сэтшöм визюв мый
32136 37646 ю вылас кö пыжöн сынан сразу пыр öтнад кö сынан öтнад он вермы мыйкö керны, сразу нуыштас кытчö кö.
38270 40203 И если кö,
41586 44076 на пример öтчыд мöдöдчим
46886 51081 Йилемъяскöд мöдöдчим пужöн льöм вотны.
51081 53745 Мунім, кыытім, кыытім, юöдіс
53745 56833 и друг сэтшöм виам воис
56833 60810 мый миянлысь пыжнымöс бергöдіс и ставным усим юас.
60810 64316 А оказывайтся абу вöлöма йир и
64316 68660 ми гортöдз эта берег дорöдзыс
68881 71573 йылан котырыс öдва и доберитчим.
72596 73573 Сэсся
75170 81305 миян колö мунны районнöй чентрсяньыс Усть-Вачерга сиктöдзыс квайтымын километр.
81305 82626 квайтымын верс.
82835 86938 И сэті туйыс зэв лёк, а ми велöдчим
87321 91741 Кослан, Косланын, помалім семилеткасö и велöдчим
91838 93505 Косланас, районнöй чентрас.
93685 95973 И мöдам вöлі и,
96600 101615 кыз шок кодь ныдын эськама туйсö кыдз абуджыка прöдитім.
101615 104728 Мöдöдча гöг-, туйыс зэв няйтöсь.
104728 109426 И пыр пöшти вöлнас оз ёна ветлö.
109426 113691 И ветлöдлам унджыкысö ветлöдлім подöн, мый öд
113906 116553 йилöмъясыдлöн миян кокным öд ён.
116553 118088 Öдйö вермам мунны.
118483 119196 Сэсся,
120585 127425 öтпыр ми мунім, мунім, да друг сэтшöм пемыд лоис и мый нинöм оз тыдав.
127425 135338 Няйтыс, няйтас вöлі, сэтшöм няйтöсь вöлі мый подöн кок вывті воö и.
696 3168 Täällä on Keski-Sysolan murre.
4155 4825 Кхм.
6656 9356 Ме чужлі Сысола районса Межадор сиктын.
9356 11291 Менö Конанова Александраöн.
11780 14796 Миян Межадор сикт сулалö Сыктыв ю дорын.
15908 19870 Ме öні висьтала видз вылын уджала öтик лун йылысь.
20703 23125 Миян колхозлöн видзьяс ылынöсь.
```

ELAN file validation

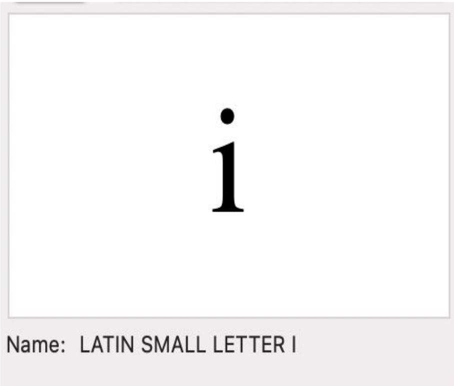
We often want to use in our transcription specific characters

We may want to transcribe all empty utterances

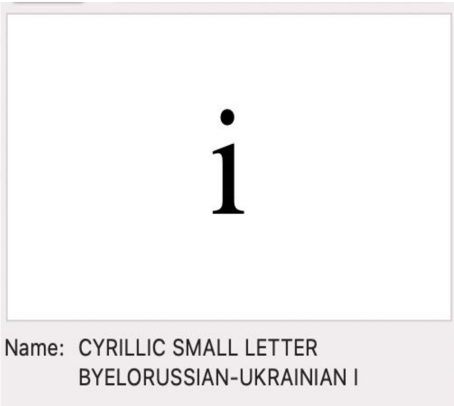
We may want to check utterances that are too long or short

In the next examples we assume the structure below

Filenames should follow a pattern, all tiers should be present



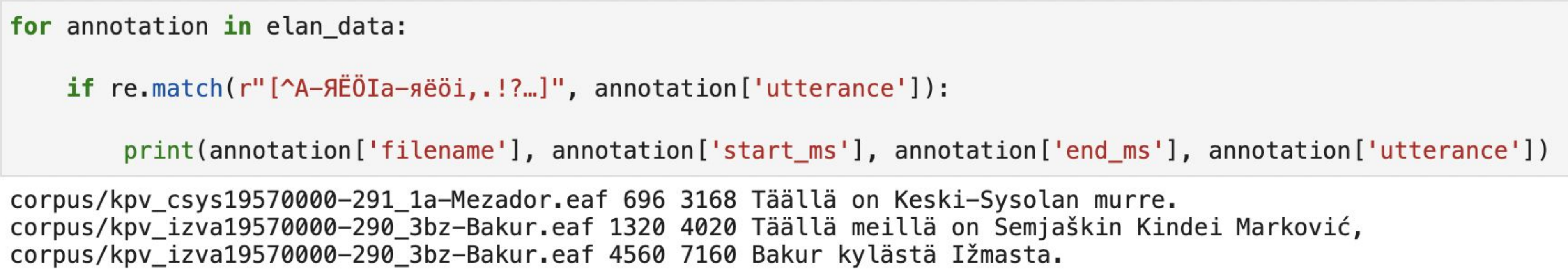
Name: LATIN SMALL LETTER I



Name: CYRILLIC SMALL LETTER BYELORUSSIAN-UKRAINIAN I

```
{'start_ms': 1765,
 'end_ms': 8825,
 'utterance': 'Ме чужи Удора районын, Усть-Вачерга сиктын, коді сулалӧ Вашка ю дорын.',
 'reference': 'kpv_udo19570000-290_3a-01',
 'participant': 'XUV-F-1920',
 'filename': 'corpus/kpv_udo19570000-290_3a-Ust-Vacerga.eaf'}
```

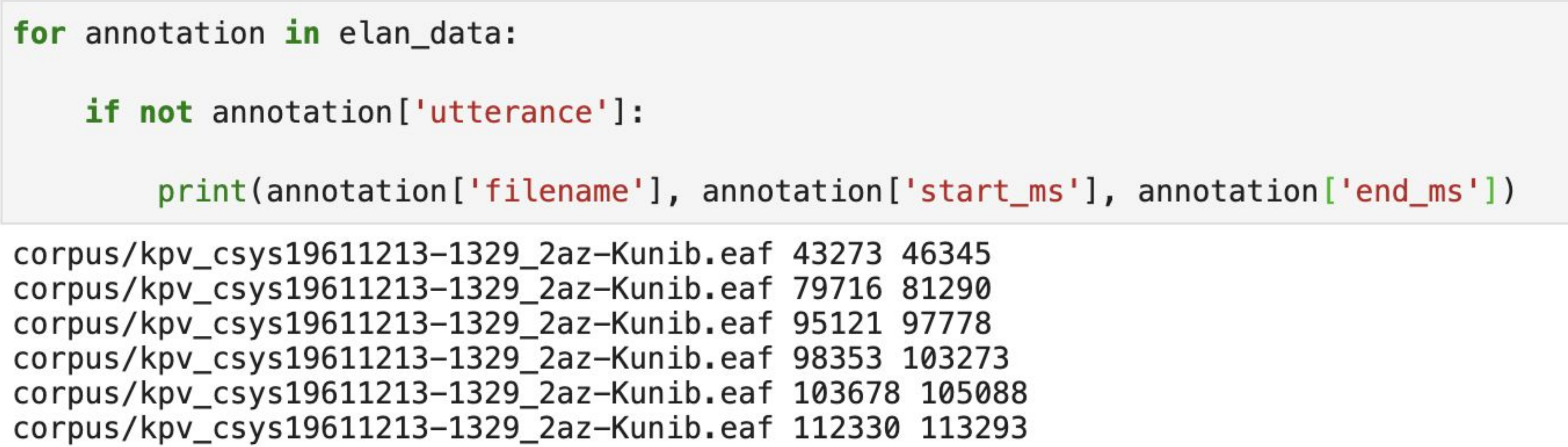
Checking for non-allowed characters



```
for annotation in elan_data:

    if re.match(r"[^A-ЯЁÖIa-яёöi,.!?…]", annotation['utterance']):

        print(annotation['filename'], annotation['start_ms'], annotation['end_ms'], annotation['utterance'])
```

```
corpus/kpv_csys19570000-291_1a-Mezador.eaf 696 3168 Täällä on Keski-Sysolan murre.
corpus/kpv_izva19570000-290_3bz-Bakur.eaf 1320 4020 Täällä meillä on Semjaškin Kindei Marković,
corpus/kpv_izva19570000-290_3bz-Bakur.eaf 4560 7160 Bakur kylästä Ižmasta.
```

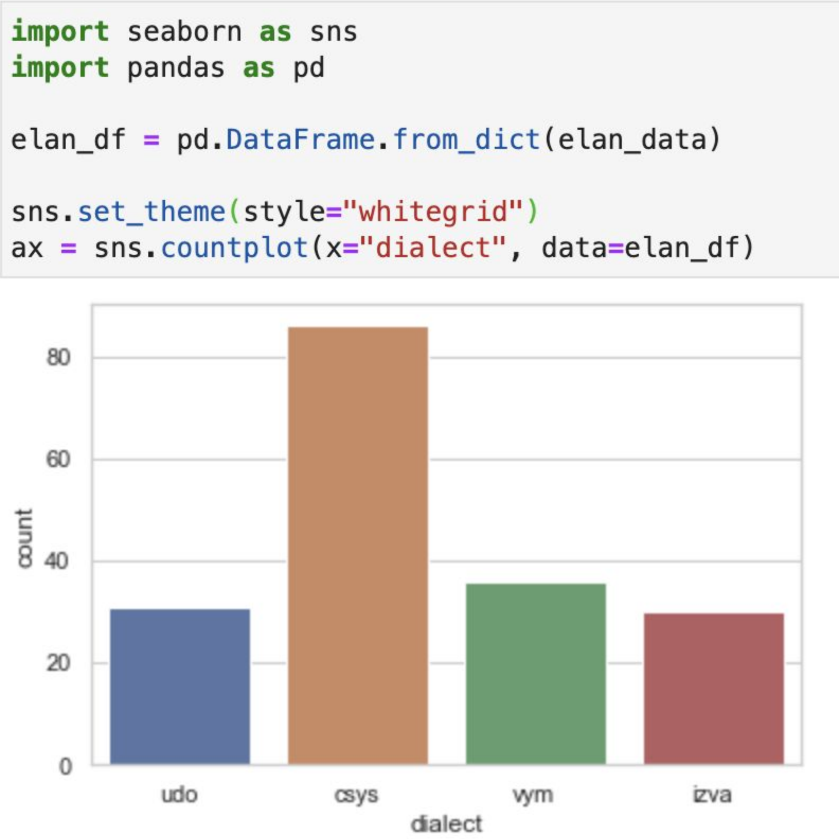
Checking for empty utterances



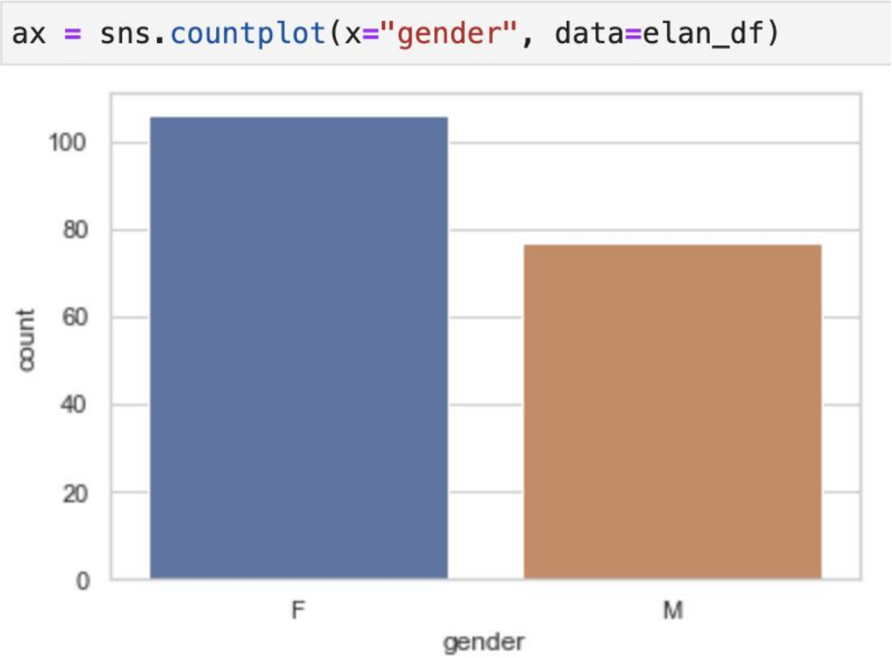
```
for annotation in elan_data:

    if not annotation['utterance']:

        print(annotation['filename'], annotation['start_ms'], annotation['end_ms'])
```

```
corpus/kpv_csys19611213-1329_2az-Kunib.eaf 43273 46345
corpus/kpv_csys19611213-1329_2az-Kunib.eaf 79716 81290
corpus/kpv_csys19611213-1329_2az-Kunib.eaf 95121 97778
corpus/kpv_csys19611213-1329_2az-Kunib.eaf 98353 103273
corpus/kpv_csys19611213-1329_2az-Kunib.eaf 103678 105088
corpus/kpv_csys19611213-1329_2az-Kunib.eaf 112330 113293
```

We can also directly start to analyze the corpus

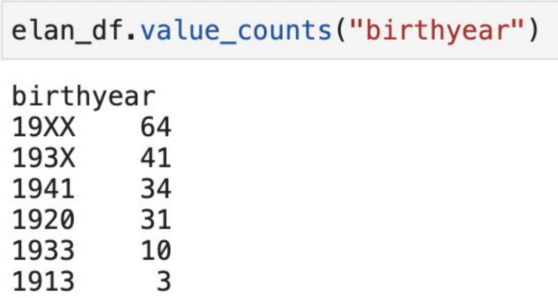


```python
import seaborn as sns
import pandas as pd

elan_df = pd.DataFrame.from_dict(elan_data)

sns.set_theme(style="whitegrid")
ax = sns.countplot(x="dialect", data=elan_df)
```



```python
ax = sns.countplot(x="gender", data=elan_df)
```

This leads to more validation questions…



```
elan_df.value_counts("birthyear")
```

```
birthyear
19XX    64
193X    41
1941    34
1920    31
1933    10
1913     3
```

Do all files have a correct naming convention?

Do all participants have a correct naming convention?

When we have more complicated metadata, there is even more to check

- Are the coordinates of locations correct?
- Is a birthyear specified for everyone, what about the recording time?

Metadata can be stored in many places: filenames, participant id’s, databases

- For analysis it doesn’t really matter where we store them – only the validity

And finally we can do very satisfying analysis!

Is variable X more common in dialect area A or B?

Is there a progressing change when recordings of different age are compared?

Everything we want to analyze depends from our data being correctly organized, and valid for the properties we want to study and inspect!

The corpus doesn’t need to be perfect!

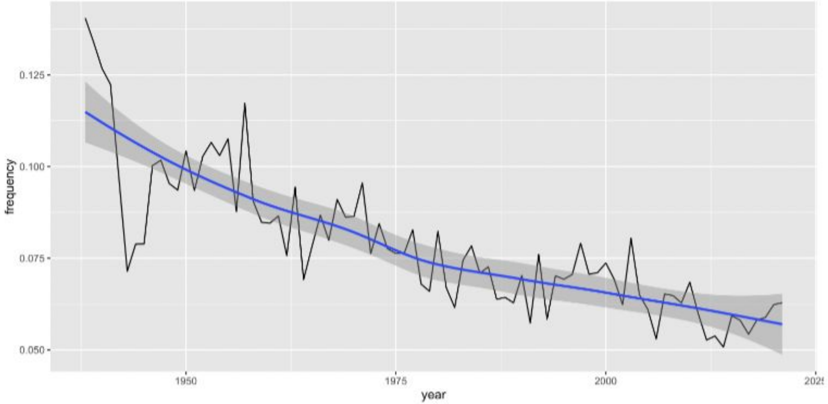


Figure 1: Relative frequency of allomorph -inɨ within third person plural past tense verbs