BELA

Le Tuan Anh <tuananh.ke@gmail.com>

Apr 12, 2022

USEFUL LINKS

1 Getting started **3**

2 Sample code **5**

2.1 BELA Tutorials 5

2.2 BELA API reference 5

2.3 BELA Changelog 8

3 Indices and tables **9**

Python Module Index **11**

Index **13**

BELA (BLIP ELAN Language Annotation) is a pathway for creating and analysing multi-lingual transcripts using BELA convention and ELAN software.

CHAPTER

ONE

GETTING STARTED

BELA is available on PyPI and can be installed using pip:

```
pip install bela
```

CHAPTER

TWO

SAMPLE CODE

The following code snippet reads a BELA transcript and prints out all participants and their utterances & chunks.

```
import bela

b2 = bela.read_eaf("my_bela_filename.eaf")
for person in b2.persons:
    print(person.name, person.code)
    for u in person.utterances:
        print(u, u.from_ts, u.to_ts, u.duration)
        if u.translation:
            print(u.translation)
        for c in u.chunks:
            print(f"  - {c} [{c.language}]")
```

2.1 BELA Tutorials

To be updated.

For BELA API reference, please visit *BELA API reference* page.

2.2 BELA API reference

For most people, *bela.read_eaf()* is the first thing to look at. This function returns a *bela.Bela2* object for manipulating a BELA transcript directly:

```
>>> import bela
>>> b2 = bela.read_eaf("my_bela_filename.eaf")
```

Now you can use the created b2 object to process BELA data.

```
>>> for person in b2.persons:
>>>     print(person.name, person.code)
>>>     for u in person.utterances:
>>>         print(u, u.from_ts, u.to_ts, u.duration)
>>>         if u.translation:
>>>             print(u.translation)
>>>         for c in u.chunks:
>>>             print(f"  - {c} [{c.language}]")
```

2.2.1 The bela module

bela.**read_eaf**(*eaf_path*, ***kwargs*)

Read an EAF file as a Bela2 object

Parameters **eaf_path** (*str-like object or a Path object*) – Path to the EAF file

Returns A Bela2 object

Return type *bela.Bela2*

bela.**from_elan**(*elan*, *eaf_path=':memory:'*, ***kwargs*)

Create a BELA-con version 2.x object from a `speach.elan.ELANDoc` object

2.2.2 The lex module

This module provides lexicon analysis functions (i.e. counting tokens, calculating class-token ratio, et cetera). New users should start with *bela.lex.CorpusLexicalAnalyser*.

```
>>> from bela.lex import CorpusLexicalAnalyser
>>> analyser = CorpusLexicalAnalyser()
>>> for person in b2.persons:
>>>     for u in person.utterances:
>>>         analyser.add(u.text, u.language, source=source, speaker=person.code)
>>> analyser.analyse()
```

class bela.lex.**CorpusLexicalAnalyser**(*filepath=':memory:'*, *lang_lex_map=None*, *word_only=False*, *lemmatizer=True*, ***kwargs*)

Analyse a corpus text

analyse(*external_tokenizer=True*)

Analyse all available profiles (i.e. speakers)

read(***kwargs*)

Read the CSV file content specified by self.filepath

to_dict()

Export analysed result as a JSON-ready object

2.2.3 BELA-con version 2.0 API

The official Bela convention. By default, this should be used for new transcripts.

class bela.**Bela2**(*elan*, *path=':memory:'*, *allow_empty=False*, *nlp_tokenizer=False*, *word_only=True*, *ellipsis=True*, *validate_baby_languages=False*, *ansi_languages=('English', 'Vocal Sounds', 'Malay', 'Red Dot', ':v:airstream', ':v:crying', ':v:vocalizations')*, *auto_tokenize=True*, *split_punc=True*, *remove_punc=True*, ***kwargs*)

BELA-convention version 2

find_turns(*threshold=1500*)

Find potential turn-takings

Parameters **threshold** (*float*) – Delay between utterances in milliseconds

Returns List of utterance pairs (2-tuple) (from utterance, to utterance object)

static from_elan(*elan*, *eaf_path=':memory:'*, ***kwargs*)

Create a BELA-con version 2.x object from a `speach.elan.ELANDoc` object

parse_name(*tier*)

(Internal) Parse participant name and tier type from a tier object and then update the tier object

This function is internal and should not be used outside of this class.

Parameters **tier** (*speach.elan.ELANTier*) – The tier object to parse

static read_eaf(*eaf_path*, ***kwargs*)

Read an EAF file as a Bela2 object

Parameters **eaf_path** (*str-like object or a Path object*) – Path to the EAF file

Returns A Bela2 object

Return type *bela.Bela2*

to_language_mix(*to_ts=None*, *auto_compute=True*)

Collapse utterances to generate a language mix timeline

tokenize()

tokenize all utterances

property annotation

Get an annotation object by ID

property participant_codes

Immutable list of participant codes

property person_map

Map participant (i.e. person code) to person object

property persons

All Person objects in this BELA object

property roots

Direct access to all underlying ELAN root tiers

2.2.4 BELA-con version 1.0 API

Bela1 is deprecated from Mar 2020. It is still available for backward compatible only. Please do not use it for anything other than BLIP's PILOT10 corpus.

class bela.**Bela1**

This class represent BELA convention version 1

static read(*filepath*, *autotag=True*)

Read ELAN csv file

to_language_mix(*to_ts=None*, *auto_compute=True*)

Collapse utterances to generate a language mix timeline

2.3 BELA Changelog

2.3.1 BELA 2.0.0a22 [WIP]

- Added tokenize() function to utterances and chunks
- Added the first working prototype of BELA builder (2022-03-29)
- Added Bela2.save() function
- Kickstarted BELA documentation
- Added BELA documentation: https://bela.readthedocs.io/
- Added ANSI & baby language checking rules
- use `speach` >= 0.1a15.post1
- Exposed `read_eaf()` and `from_elan()` to module level
- Exposed `media_file`, `media_url`, `relative_media_url` properties
- Fixed None utterances & chunks for not well-formed transcripts

2.3.2 BELA 2.0.0a21

- Use `speach` > 0.1a14 to support Python 3.10 and 3.11
- Updated annotation mapping mechanism
- Warn users if OMW-1.4 dataset is missing
- Clean up ~ characters after plus-to-space expansion

2.3.3 BELA 2.0.0a19

- 2022-01-26
  - Released bela-2.0.0a19 to PyPI: https://pypi.org/project/bela/2.0.0a19/

CHAPTER

THREE

INDICES AND TABLES

- genindex
- modindex
- search

PYTHON MODULE INDEX

b

bela, 6
bela.lex, 6

INDEX

A

analyse() (*bela.lex.CorpusLexicalAnalyser method*), 6
annotation (*bela.Bela2 property*), 7

B

bela
    module, 6
bela.lex
    module, 6
Bela1 (*class in bela*), 7
Bela2 (*class in bela*), 6

C

CorpusLexicalAnalyser (*class in bela.lex*), 6

F

find_turns() (*bela.Bela2 method*), 6
from_elan() (*bela.Bela2 static method*), 6
from_elan() (*in module bela*), 6

M

module
    bela, 6
    bela.lex, 6

P

parse_name() (*bela.Bela2 method*), 7
participant_codes (*bela.Bela2 property*), 7
person_map (*bela.Bela2 property*), 7
persons (*bela.Bela2 property*), 7

R

read() (*bela.Bela1 static method*), 7
read() (*bela.lex.CorpusLexicalAnalyser method*), 6
read_eaf() (*bela.Bela2 static method*), 7
read_eaf() (*in module bela*), 6
roots (*bela.Bela2 property*), 7

T

to_dict() (*bela.lex.CorpusLexicalAnalyser method*), 6
to_language_mix() (*bela.Bela1 method*), 7
to_language_mix() (*bela.Bela2 method*), 7
tokenize() (*bela.Bela2 method*), 7