wikt2pron Documentation

Release 0.0.2

Yifan Xiong

Jul 05, 2019

Contents

Wiktionary pronunciation collector

A Python toolkit converting pronunciation in enwiktionary xml dump to cmudict format. Support IPA and X-SAMPA format at present.

This project is developed in GSoC 2017 with CMU Sphinx.

Blogs for this project can be found at my Blogspot.

Collected pronunciation dictionaries and related example models can be downloaded at Dropbox.

CHAPTER 1

Contents

1.1 Introduction

wikt2pron is a Python toolkit converting pronunciation in enwiktionary xml dump to cmudict format. Support IPA and X-SAMPA format at present.

1.1.1 Features

- Extract pronunciation from Wiktionary XML dump.
- Lookup pronunciation for a word in Wiktionary.
- IPA -> X-SAMPA conversion.

1.1.2 Requirements

wikt2pron requires:

- Python 3
- regex
- python-mwxml
- beautifulsoup4

1.1.3 Installation

```
# download the latest version
$ git clone https://github.com/abuccts/wikt2pron.git
$ cd enwiktionary

# install and run test
$ python setup.py install
$ python setup.py -q test

# make documents
$ make -C docs html
```

1.2 Usage

1.2.1 Extract pronunciation from Wiktionary XML dump

First, create an instance of `Wiktionary` class:

```
>>> from pywiktionary import Wiktionary
>>> wikt = Wiktionary(XSAMPA=True)
```

Use the example XML dump in `pywiktionary/data`:

```
>>> dump_file = "pywiktionary/data/enwiktionary-test-pages-articles-
↪multistream.xml"
>>> pron = wikt.extract_IPA(dump_file)
```

Here's the extracted result:

```
>>> from pprint import pprint
>>> pprint(pron)
[{'id': 16,
  'pronunciation': {'English': [{'IPA': '/dk()n()/',
                                 'X-SAMPA': '/"dIkS(@)n(@)r\\I/',
                                 'lang': 'en'},
                                {'IPA': '/dkni/',
                                 'X-SAMPA': '/"dIkS@nEr\\i/',
                                 'lang': 'en'}]},
  'title': 'dictionary'},
 {'id': 65195,
  'pronunciation': {'English': 'IPA not found.'},
  'title': 'battleship'},
 {'id': 39478,
  'pronunciation': {'English': [{'IPA': '/md()/',
                                 'X-SAMPA': '/"m3:d@(r\\)/',
                                 'lang': 'en'},
                                {'IPA': '/m.d/',
                                 'X-SAMPA': '/"m3`.d@`/',
                                 'lang': 'en'}]},
  'title': 'murder'},
 {'id': 80141,
  'pronunciation': {'English': [{'IPA': '/dæzl/',
                                 'X-SAMPA': '/"d{z@l/',
```

(continues on next page)

(continued from previous page)

```
                                    'lang': 'en'}]},
 'title': 'dazzle'}]
```

1.2.2 Lookup pronunciation for a word in Wiktionary

First, create an instance of `Wiktionary` class:

```
>>> from pywiktionary import Wiktionary
>>> wikt = Wiktionary(XSAMPA=True)
```

Lookup a word using `lookup` method:

```
>>> word = wikt.lookup("present")
```

The entry of word "present" is at https://en.wiktionary.org/wiki/present, and here is the lookup result:

```
>>> from pprint import pprint
>>> pprint(word)
{'Catalan': 'IPA not found.',
 'Danish': [{'IPA': '/prsan/', 'X-SAMPA': '/prEsang/', 'lang': 'da'},
            {'IPA': '[ps■]', 'X-SAMPA': '[p_hR_0E"sAN]', 'lang': 'da'}
],
 'English': [{'IPA': '/pznt/', 'X-SAMPA': '/"pr\\Ez@nt/', 'lang': 'en'},
             {'IPA': '/pznt/', 'X-SAMPA': '/pr\\I"zEnt/', 'lang': 'en'},
             {'IPA': '/pznt/', 'X-SAMPA': '/pr\\@"zEnt/', 'lang': 'en'}],
 'Ladin': 'IPA not found.',
 'Middle French': 'IPA not found.',
 'Old French': 'IPA not found.',
 'Swedish': [{'IPA': '/present/', 'X-SAMPA': '/pre"sent/', 'lang': 'sv'}]}
```

To lookup a word in a certain language, specify the `lang` parameter:

```
>>> wikt = Wiktionary(lang="English", XSAMPA=True)
>>> word = wikt.lookup("read")
>>> pprint(word)
[{'IPA': '/id/', 'X-SAMPA': '/r\\i:d/', 'lang': 'en'},
 {'IPA': '/d/', 'X-SAMPA': '/r\\Ed/', 'lang': 'en'}]
```

1.2.3 IPA -> X-SAMPA conversion

```
>>> from pywiktionary import IPA
>>> IPA_text = "/tend/" # en: [[change]]
>>> XSAMPA_text = IPA.IPA_to_XSAMPA(IPA_text)
>>> XSAMPA_text
"/t__SeInd__Z/"
```

1.2.4 Using the collected dictionaries

To use the collected dictionaries training G2P models or acoustic models, please refer to these blogs for details:

1. Grapheme to Phoneme Conversion

2. Training Acoustic Model on Voxforge Dataset
3. Training Acoustic Model on LibriSpeech

1.3 pywiktionary API

The library provides classes which are usable by third party tools.

1.3.1 `Wiktionary` Class

class pywiktionary.**Wiktionary**(*lang=None*, *XSAMPA=False*)

Wiktionary class for IPA extraction from XML dump or MediaWiki API.

To extraction IPA for a certain language, specify `lang` parameter, default is extracting IPA for all available languages.

To convert IPA text to X-SAMPA text, use `XSAMPA` parameter.

Parameters

- **lang** (*string*) – String of language type.
- **XSAMPA** (*boolean*) – Option for IPA to X-SAMPA conversion.

extract_IPA(*dump_file*)

Extraction IPA list from Wiktionary XML dump.

Parameters **dump_file** (*string*) – Path of Wiktionary XML dump file.

Returns List of extracted IPA results in `{"id": "", "title": "", "pronunciation": ""}` format.

Return type list

get_entry_pronunciation(*wiki_text*, *title=None*)

Extraction IPA for entry in Wiktionary XML dump.

Parameters

- **wiki_text** (*string*) – String of XML entry wiki text.
- **title** (*string*) – String of wiki entry title.

Returns Dict of word's IPA results. Key: language name; Value: list of IPA text.

Return type dict

lookup(*word*)

Look up IPA of word through Wiktionary API.

Parameters **word** (*string*) – String of a word to be looked up.

Returns Dict of word's IPA results. Key: language name; Value: list of IPA text.

Return type dict

set_XSAMPA(*XSAMPA*)
Set X-SAMPA conversion option.

Parameters **XSAMPA** (*boolean*) – Option for IPA to X-SAMPA conversion.

set_lang(*lang*)
Set language.

Parameters **lang** (*string*) – String of language name.

set_parser()
Set parser for Wiktionary.

Use the Wiktionary `lang` and `XSAMPA` parameters.

1.3.2 `Parser` Class

class pywiktionary.**Parser**(*lang=None*, *XSAMPA=False*)
Wiktionary parser to extract IPA text from pronunciation section.

To extraction IPA for a certain language, specify `lang` parameter, default is extracting IPA for all available languages.

To convert IPA text to X-SAMPA text, use `XSAMPA` parameter.

Parameters

- **lang** (*string*) – String of language type.
- **XSAMPA** (*boolean*) – Option for IPA to X-SAMPA conversion.

expand_template(*text*)
Expand IPA Template through Wiktionary API.

Used to expand `{{*-IPA}}` template in parser and return IPA list.

Parameters **text** (*string*) – String of template text inside "{{" and "}}".

Returns List of expanded IPA text.

Return type list of string

Examples

```
>>> parser = Parser()
>>> template = "{{la-IPA|eccl=yes|thēsaurus}}"
>>> parser.expand_template(template)
['/tesau.rus/', '[tesau.rs]', '/tesau.rus/']
```

parse(*wiki_text*, *title=None*)
Parse Wiktionary wiki text.

Split Wiktionary wiki text into different langugaes and return parseed IPA result.

Parameters

- **wiki_text** (*string*) – String of Wiktionary wiki text, from XML dump or Wiktionary API.
- **title** (*string*) – String of wiki entry title.

Returns Dict of parsed IPA results. Key: language name; Value: list of IPA text.

Return type dict

parse_detail(*wiki_text*, *depth=3*)
Parse the section of a certain language in wiki text.

Parse pronunciation section of the certain language recursively.

Parameters

- **wiki_text** (*string*) – String of wiki text in a language section.
- **depth** (*int*) – Integer indicated depth of pronunciation section.

Returns List of extracted IPA text in `{"IPA": "", "X-SAMPA": "", "lang": ""}` format.

Return type list of dict

parse_pronunciation(*wiki_text*)
Parse pronunciation section in wiki text.

Parse IPA text from pronunciation section and convert to X-SAMPA.

Parameters **wiki_text** (*string*) – String of pronunciation section in wiki text.

Returns List of extracted IPA text in `{"IPA": "", "X-SAMPA": "", "lang": ""}` format.

Return type list of dict

1.3.3 Utilities

IPA and X-SAMPA related variables and functions. Modified from https://en.wiktionary.org/wiki/Module:IPA Lua module partially.

IPA.IPA.**IPA_to_CMUBET**(*text*)
Convert IPA to CMUBET for US English.

Use IPA and symbol set used in Wiktionary and CMUBET symbol set used in CMUDict.

Parameters **text** (*string*) – String of IPA text parsed from Wiktionary.

Returns Converted CMUBET text.

Return type string

IPA.IPA.**IPA_to_XSAMPA**(*text*)
Convert IPA to X-SAMPA.

Use IPA and X-SAMPA symbol sets used in Wiktionary.

Parameters **text** (*string*) – String of IPA text parsed from Wiktionary.

Returns Converted X-SAMPA text.

Return type string

Notes

- Use _j for palatalized instead of '
- Use = for syllabic instead of _=
- Use ~ for nasalization instead of _~

- Please refer to *IPA <-> X-SAMPA Symbol Set* for more details.

Examples

```
>>> IPA_text = "/tend/" # en: [[change]]
>>> XSAMPA_text = IPA_to_XSAMPA(IPA_text)
>>> XSAMPA_text
"/t__SeInd__Z/"
```

Convert spelling text in {{*-IPA}} to IPA pronunciation.

Most are modified from Wiktionary Lua Module.

IPA.fr_pron.**to_IPA**(*text*, *pos=''*)

Generates French IPA from spelling.

Implements template {{fr-IPA}}.

Parameters

- **text** (*string*) – String of fr-IPA text parsed in {{fr-IPA}} from Wiktionary.
- **pos** (*string*) – String of |pos= parameter parsed in {{fr-IPA}}.

Returns Converted French IPA.

Return type string

Notes

- Modified from Wiktioanry fr-pron Lua module partially.
- Rewritten from rewritten by *Benwing* and original by *Kc kennylau*.
- Testcases are modified from Wiktionary fr-pron/testcases.

Examples

```
>>> fr_text = "hæmorrhagie" # fr: [[hæmorrhagie]]
>>> fr_IPA = fr_pron.to_IPA(fr_text)
>>> fr_IPA
"e.m.a.i"
```

IPA.ru_pron.**to_IPA**(*text*, *adj=''*, *gem=''*, *bracket=''*, *pos=''*)

Generates Russian IPA from spelling.

Implements template {{ru-IPA}}.

Parameters

- **text** (*string*) – String of ru-IPA text parsed in {{ru-IPA}} from Wiktionary.
- **adj** (*string*) – String of |noadj= parameter parsed in {{ru-IPA}}.
- **gem** (*string*) – String of |gem= parameter parsed in {{ru-IPA}}.
- **bracket** (*string*) – String of |bracket= parameter parsed in {{ru-IPA}}.
- **pos** (*string*) – String of |pos= parameter parsed in {{ru-IPA}}.

Returns Converted Russian IPA.

Return type string

Notes

- Modified from Wiktioanry ru-pron Lua module partially.
- Rewritten from Author: Originally *Wyang*; rewritten by *Benwing*; additional contributions from *Atitarev* and a bit from others.
- Testcases are modified from Wiktionary ru-pron/testcases.

Examples

```
>>> ru_text = "" # ru: [[]]
>>> ru_IPA = ru_pron.to_IPA(ru_text)
>>> ru_IPA
"slivj"
```

IPA.hi_pron.**to_IPA**(*text*)

Generates Hindi IPA from spelling.

Implements template {{hi-IPA}}.

Parameters **text** (*string*) – String of hi-IPA text parsed in {{hi-IPA}} from Wiktionary.

Returns Converted Hindi IPA.

Return type string

Notes

- Modified from Wiktioanry hi-IPA Lua module partially.
- Testcases are modified from Wiktionary hi-IPA/testcases.

Examples

```
>>> hi_text = "" # hi: [[]]
>>> hi_IPA = hi_pron.to_IPA(hi_text)
>>> hi_IPA
"m"
```

IPA.es_pron.**to_IPA**(*word*, *LatinAmerica=False*, *phonetic=True*)

Generates Spanish IPA from spelling.

Implements template {{es-IPA}}.

Parameters

- **word** (*string*) – String of es-IPA text parsed in {{es-IPA}} from Wiktionary.
- **LatinAmerica** (*bool*) – Value of |LatinAmerica= parameter parsed in {{es-IPA}}.
- **phonetic** (*bool*) – Value of |phonetic= parameter parsed in {{es-IPA}}.

Returns Converted Spanish IPA.

Return type string

Notes

- Modified from Wiktioanry es-pronunc Lua module partially.
- Testcases are modified from Wiktionary es-pronunc/testcases.

Examples

```
>>> es_text = "baca" # es: [[baca]]
>>> es_IPA = es_pron.to_IPA(es_text)
>>> es_IPA
"baka"
```

IPA.cmn_pron.**to_IPA**(*text*, *IPA_tone=True*)

Generates Mandarin IPA from Pinyin.

Implements `|m=` parameter for template {{zh-pron}}.

Parameters

- **text** (*string*) – String of `|m=` parameter parsed in {{zh-pron}} from Wiktionary.
- **IPA_tone** (*bool*) – Whether add IPA tone in result.

Returns Converted Mandarin IPA.

Return type string

Notes

- Modified from Wiktioanry cmn-pron Lua module partially.

Examples

```
>>> cmn_text = "pīnyīn" # zh: [[]]
>>> cmn_IPA = cmn_pron.to_IPA(cmn_text)
>>> cmn_IPA
"pin55 in55"
```

1.4 IPA <-> X-SAMPA Symbol Set

```
# X-SAMPA symbols
data = {
    # not in official X-SAMPA; from http://www.kneequickie.com/kq/Z-SAMPA
    "b\\": {
        "IPA_symbol": "",
    },
    "b_<": {
        "IPA_symbol": "",
    },
    "d`": {
        "IPA_symbol": "",
        "has_descender": True,
```

(continues on next page)

(continued from previous page)

```
    },
    "d_<": {
        "IPA_symbol": "",
    },
    # not in official X-SAMPA; Wikipedia-specific
    "d`_<": {
        "IPA_symbol": "",
        "has_descender": True,
    },
    "g": {
        "IPA_symbol": "",
        "has_descender": True,
    },
    "g_<": {
        "IPA_symbol": "",
        "has_descender": True,
    },
    "h\\": {
        "IPA_symbol": "",
    },
    "j\\": {
        "IPA_symbol": "",
        "has_descender": True,
    },
    "l`": {
        "IPA_symbol": "",
        "has_descender": True,
    },
    "l\\": {
        "IPA_symbol": "",
    },
    "n`": {
        "IPA_symbol": "",
        "has_descender": True,
    },
    "p\\": {
        "IPA_symbol": "",
        "has_descender": True,
    },
    "r`": {
        "IPA_symbol": "",
        "has_descender": True,
    },
    "r\\": {
        "IPA_symbol": "",
    },
    "r\\`": {
        "IPA_symbol": "",
        "has_descender": True,
    },
    "s`": {
        "IPA_symbol": "",
        "has_descender": True,
    },
    "s\\": {
        "IPA_symbol": "",
    },
```

(continues on next page)

(continued from previous page)

```
    "t`": {
        "IPA_symbol": "",
    },
    "v\\": {
        "IPA_symbol": "",
    },
    "x\\": {
        "IPA_symbol": "",
        "has_descender": True,
    },
    "z`": {
        "IPA_symbol": "",
        "has_descender": True,
    },
    "z\\": {
        "IPA_symbol": "",
    },
    "A": {
        "IPA_symbol": "",
    },
    "B": {
        "IPA_symbol": "β",
        "has_descender": True,
    },
    "B\\": {
        "IPA_symbol": "",
    },
    "C": {
        "IPA_symbol": "ç",
        "has_descender": True,
    },
    "D": {
        "IPA_symbol": "ð",
    },
    "E": {
        "IPA_symbol": "",
    },
    "F": {
        "IPA_symbol": "",
        "has_descender": True,
    },
    "G": {
        "IPA_symbol": "",
        "has_descender": True,
    },
    "G\\": {
        "IPA_symbol": "",
    },
    "G\\_<": {
        "IPA_symbol": "",
    },
    "H": {
        "IPA_symbol": "",
        "has_descender": True,
    },
    "H\\": {
        "IPA_symbol": "",
```

(continues on next page)

(continued from previous page)


```
    },
    "I": {
        "IPA_symbol": "",
    },
    "I\\": {
        "IPA_symbol": "",
    },
    "J": {
        "IPA_symbol": "",
        "has_descender": True,
    },
    "J\\": {
        "IPA_symbol": "",
    },
    "J\\_<": {
        "IPA_symbol": "",
        "has_descender": True,
    },
    "K": {
        "IPA_symbol": "",
    },
    "K\\": {
        "IPA_symbol": "",
        "has_descender": True,
    },
    "L": {
        "IPA_symbol": "",
    },
    "L\\": {
        "IPA_symbol": "",
    },
    "M": {
        "IPA_symbol": "",
    },
    "M\\": {
        "IPA_symbol": "",
        "has_descender": True,
    },
    "N": {
        "IPA_symbol": "■",
        "has_descender": True,
    },
    "N\\": {
        "IPA_symbol": "",
    },
    "O": {
        "IPA_symbol": "",
    },
    "O\\": {
        "IPA_symbol": "",
    },
    "P": {
        "IPA_symbol": "",
    },
    "Q": {
        "IPA_symbol": "",
    },
```


(continues on next page)

(continued from previous page)

```
    "R": {
        "IPA_symbol": "",
    },
    "R\\": {
        "IPA_symbol": "",
    },
    "S": {
        "IPA_symbol": "",
        "has_descender": True,
    },
    "T": {
        "IPA_symbol": "θ",
    },
    "U": {
        "IPA_symbol": "",
    },
    "U\\": {
        "IPA_symbol": "",
    },
    "V": {
        "IPA_symbol": "",
    },
    "W": {
        "IPA_symbol": "",
    },
    "X": {
        "IPA_symbol": "χ",
        "has_descender": True,
    },
    "X\\": {
        "IPA_symbol": "",
    },
    "Y": {
        "IPA_symbol": "",
    },
    "Z": {
        "IPA_symbol": "",
        "has_descender": True,
    },
    "\"": {
        "IPA_symbol": "",
    },
    "%": {
        "IPA_symbol": "",
    },
    # not in official X-SAMPA; from http://www.kneequickie.com/kq/Z-SAMPA
    "%\\": {
        "IPA_symbol": "",
    },
    "'": {
        "IPA_symbol": "",
        "is_diacritic": True,
    },
    ":": {
        "IPA_symbol": "",
        "is_diacritic": True,
    },
```

(continues on next page)

(continued from previous page)


```
    ":\\": {
        "IPA_symbol": "",
        "is_diacritic": True,
    },
    "@": {
        "IPA_symbol": "",
    },
    "@`": {
        "IPA_symbol": "",
    },
    "@\\": {
        "IPA_symbol": "",
    },
    "{": {
        "IPA_symbol": "æ",
    },
    "}": {
        "IPA_symbol": "",
    },
    "1": {
        "IPA_symbol": "",
    },
    "2": {
        "IPA_symbol": "ø",
    },
    "3": {
        "IPA_symbol": "",
    },
    "3`": {
        "IPA_symbol": "",
    },
    "3\\": {
        "IPA_symbol": "",
    },
    "4": {
        "IPA_symbol": "",
    },
    "5": {
        "IPA_symbol": "",
    },
    "6": {
        "IPA_symbol": "",
    },
    "7": {
        "IPA_symbol": "",
    },
    "8": {
        "IPA_symbol": "",
    },
    "9": {
        "IPA_symbol": "œ",
    },
    "&": {
        "IPA_symbol": "",
    },
    "?": {
        "IPA_symbol": "",
```


(continues on next page)

(continued from previous page)

```
    },
    "?\\": {
        "IPA_symbol": "",
    },
    "<\\": {
        "IPA_symbol": "",
    },
    ">\\": {
        "IPA_symbol": "",
    },
    "^": {
        "IPA_symbol": "",
    },
    "!": {
        "IPA_symbol": "",
    },
    # not in official X-SAMPA
    "!!": {
        "IPA_symbol": "",
    },
    "!\\": {
        "IPA_symbol": "",
    },
    "|\\": {
        "IPA_symbol": "",
        "has_descender": True,
    },
    "||": {
        "IPA_symbol": "||",
        "has_descender": True,
    },
    "|\\|\\": {
        "IPA_symbol": "",
        "has_descender": True,
    },
    "=\\": {
        "IPA_symbol": "",
        "has_descender": True,
    },
    # linking mark, liaison
    "-\\": {
        "IPA_symbol": "",
        "is_diacritic": True,
    },
    # coarticulated; not in official X-SAMPA
    "__": {
        "IPA_symbol": u"\u0361",
    },
    # fortis, strong articulation; not in official X-SAMPA
    "_:": {
        "IPA_symbol": u"\u0348",
    },
    "_\"": {
        "IPA_symbol": u"\u0308",
        "is_diacritic": True,
    },
    # advanced
```

(continues on next page)

(continued from previous page)

```
    "_+": {
        "IPA_symbol": u"\u031F",
        "with_descender": "",
        "is_diacritic": True,
    },
    # retracted
    "_-": {
        "IPA_symbol": u"\u0320",
        "with_descender": "",
        "is_diacritic": True,
    },
    # rising tone
    "_/": {
        "IPA_symbol": u"\u030C",
        "is_diacritic": True,
    },
    # voiceless
    "_0": {
        "IPA_symbol": u"\u0325",
        "with_descender": u"\u030A",
        "is_diacritic": True,
    },
    # syllabic
    "=": {
        "IPA_symbol": u"\u0329",
        "with_descender": u"\u030D",
        "is_diacritic": True,
    },
    # syllabic (both are OK according to https://en.wikipedia.org/wiki/X-SAMPA)
    "_=": {
        "IPA_symbol": u"\u0329",
        "with_descender": u"\u030D",
        "is_diacritic": True,
    },
    # strident: not in official X-SAMPA;
    # from http://www.kneequickie.com/kq/Z-SAMPA
    "_%\\": {
        "IPA_symbol": u"\u1DFD",
    },
    # ejective
    "_>": {
        "IPA_symbol": "",
        "is_diacritic": True,
    },
    # pharyngealized
    "_?\\": {
        "IPA_symbol": "",
        "is_diacritic": True,
    },
    # falling tone
    "_\\": {
        "IPA_symbol": u"\u0302",
        "is_diacritic": True,
    },
    # non-syllabic
    "_^": {
        "IPA_symbol": u"\u032F",
```

(continues on next page)

(continued from previous page)

```
        "with_descender": u"\u0311",
        "is_diacritic": True,
    },
    # no audible release
    "_}": {
        "IPA_symbol": u"\u031A",
        "is_diacritic": True,
    },
    # r-coloring (colouring), rhotacization
    "`": {
        "IPA_symbol": u"\u02DE",
        "is_diacritic": True,
    },
    # nasalization
    "~": {
        "IPA_symbol": u"\u0303",
        "is_diacritic": True,
    },
    # advanced tongue root
    "_A": {
        "IPA_symbol": u"\u0318",
        "is_diacritic": True,
    },
    # apical
    "_a": {
        "IPA_symbol": u"\u033A",
        "is_diacritic": True,
    },
    # extra-low tone
    "_B": {
        "IPA_symbol": u"\u030F",
        "is_diacritic": True,
    },
    # low rising tone
    "_B_L": {
        "IPA_symbol": u"\u1DC5",
        "is_diacritic": True,
    },
    # less rounded
    "_c": {
        "IPA_symbol": u"\u031C",
        "is_diacritic": True,
    },
    # dental
    "_d": {
        "IPA_symbol": u"\u032A",
        "is_diacritic": True,
    },
    # velarized or pharyngealized (dark)
    "_e": {
        "IPA_symbol": u"\u0334",
        "is_diacritic": True,
    },
    # downstep
    "<F>": {
        "IPA_symbol": "",
    },
```

(continues on next page)

(continued from previous page)

```
    # falling tone
    "_F": {
        "IPA_symbol": u"\u0302",
        "is_diacritic": True,
    },
    # velarized
    "_G": {
        "IPA_symbol": "",
        "is_diacritic": True,
    },
    # high tone
    "_H": {
        "IPA_symbol": u"\u0301",
        "is_diacritic": True,
    },
    # high rising tone
    "_H_T": {
        "IPA_symbol": u"\u1DC4",
        "is_diacritic": True,
    },
    # aspiration
    "_h": {
        "IPA_symbol": "",
        "is_diacritic": True,
    },
    # palatalization
    "_j": {
        "IPA_symbol": "",
        "is_diacritic": True,
    },
    # creaky voice, laryngealization, vocal fry
    "_k": {
        "IPA_symbol": u"\u0330",
        "is_diacritic": True,
    },
    # low tone
    "_L": {
        "IPA_symbol": u"\u0300",
        "is_diacritic": True,
    },
    # lateral release
    "_l": {
        "IPA_symbol": "",
        "is_diacritic": True,
    },
    # mid tone
    "_M": {
        "IPA_symbol": u"\u0304",
        "is_diacritic": True,
    },
    # laminal
    "_m": {
        "IPA_symbol": u"\u033B",
        "is_diacritic": True,
    },
    # linguolabial
    "_N": {
```

(continues on next page)

(continued from previous page)

```
        "IPA_symbol": u"\u033C",
        "is_diacritic": True,
    },
    # nasal release
    "_n": {
        "IPA_symbol": "",
        "is_diacritic": True,
    },
    # more rounded
    "_O": {
        "IPA_symbol": u"\u0339",
        "is_diacritic": True,
    },
    # lowered
    "_o": {
        "IPA_symbol": u"\u031E",
        "with_descender": "",
        "is_diacritic": True,
    },
    # retracted tongue root
    "_q": {
        "IPA_symbol": u"\u0319",
        "is_diacritic": True,
    },
    # global rise
    "<R>": {
        "IPA_symbol": "",
    },
    # rising tone
    "_R": {
        "IPA_symbol": u"\u030C",
        "is_diacritic": True,
    },
    # rising falling tone
    "_R_F": {
        "IPA_symbol": u"\u1DC8",
        "is_diacritic": True,
    },
    # raised
    "_r": {
        "IPA_symbol": u"\u031D",
        "is_diacritic": True,
    },
    # extra-high tone
    "_T": {
        "IPA_symbol": u"\u030B",
        "is_diacritic": True,
    },
    # breathy voice, murmured voice, murmur, whispery voice
    "_t": {
        "IPA_symbol": u"\u0324",
        "is_diacritic": True,
    },
    # voiced
    "_v": {
        "IPA_symbol": u"\u032C",
        "is_diacritic": True,
```

(continues on next page)

(continued from previous page)

```
    },
    # labialized
    "_w": {
        "IPA_symbol": "",
        "is_diacritic": True,
    },
    # extra-short
    "_X": {
        "IPA_symbol": u"\u0306",
        "is_diacritic": True,
    },
    # mid-centralized
    "_x": {
        "IPA_symbol": u"\u033D",
        "is_diacritic": True,
    },
    "__T": {
        "IPA_symbol": "",
    },
    "__H": {
        "IPA_symbol": "",
    },
    "__M": {
        "IPA_symbol": "",
    },
    "__L": {
        "IPA_symbol": "",
    },
    "__B": {
        "IPA_symbol": "",
    },
    # not X-SAMPA; for convenience
    # dotted circle
    "0": {
        "IPA_symbol": "",
    },
}

identical = "acehklmnorstuvwxz"
for char in identical:
    data[char] = {"IPA_symbol": char}

identical_with_descender = "jpqy"
for char in identical_with_descender:
    data[char] = {"IPA_symbol": char, "has_descender": True}
```

CHAPTER 2

Authors

- Yifan Xiong – https://github.com/abuccts

CHAPTER 3

Indices and tables

- genindex
- modindex
- search

Python Module Index

i

p

Index