
kyoto-reader Documentation

Kurohashi-Kawahara Lab

Apr 11, 2022

Contents:

1	About	1
2	Requirements	3
3	Install kyoto-reader	5
4	A Brief Explanation of KWDL C and other corpora	7
5	Usage	9
6	CLI Interfaces	11
6.1	Browsing files	11
6.2	Processing Corpus	11
6.3	Zsh Completions	12
7	Documents	13
7.1	kyoto_reader package	13
8	Author/Contact	25
9	Indices and Tables	27
	Python Module Index	29
	Index	31

CHAPTER 1

About

äzhnéČ;äd'gā■ēāNāĖñéŮNāAŮāAēāAĎāCÑēfřēIdeāĖægNéĀāāCĎāĖsāRĆçĖgēŮcāfCāAŮNāzŸāyŎāAŮāCŇāAšāCšāČijaČSāCzāČŠ
āAŮNāCL'æL'sāAĖāAšāCāāAōāCđ'āČšāCĚāČijaČŤāCgāČijaČzāČSæRRā;ZāAŮāA;āAZāĀC
āAšāAōāCĎāČijaČñāAr pyknp āAōāČl'āČČāČSāČijaAģāAČāCŇāAšāCāāĀā;ćæĖŇçť'æČĖāāšāCĎāfCāCLāRŮāAšēŮcāfCāAĭāA

Table 1: āĹl'çŤlāRrèČ;āAĭāCšāČijaČSāCzāŸĀèg

Name	Domain	Size
äzhnéČ;äd'gā■ēāCēāCgāČŮæŮGæZŷāČĭāČijaČL'āCšāČijaČSāCzāČŠāCzāČL (KWDLCL)	æŮrèAđēĲāžNāČzçd'ĭèñ	16,038 æŮĜ
äzhnéČ;äd'gā■ēāCĖāC■āCzāČLāCšāČijaČSāCz (Ky-otoCorpus)	æŮrèAđēĲāžNāČzçd'ĭèñ	15,872 æŮĜ
äŷ■æžĀēĲæšzāCĜāČijaCĚāCzāČČāČLāCĚāCrazŸāA■āCšāCzāČSāCzŮcāAZāCŇāĖĲçĭf (AnnotatedFKCCorpus)	æŮrèAđēĲāžNāČzçd'ĭèñ	1,282 æŮĜ

CHAPTER 2

Requirements

- **Python**
 - Verified Versions: 3.7, 3.8, 3.9, 3.10
- pyknp 0.4.6+
- KNP (optional)
- JumanDIC (optional)

CHAPTER 3

Install kyoto-reader

[REDACTED]

or

[REDACTED]

A Brief Explanation of KWDLC and other corpora

KWDLC, KyotoCorpus, AnnotatedFKCCorpus

ãAřãAĐãAŽãCÑãCCæÛëæIñèIdãAõæÛĞæŽÿãAñãř;ãAÛãAęã;ćæĚÑçťããCĐægÑæÛĞæĈĚããšãAõãžÛãĀAęĚřèIdéãĚægÑéĀãCĐã

KWDLC ãAřãCęãCğãČÛãAÑãCL'æŁ;ãĠžãAÛãA§ 3 æÛĞãCŠ 1 æÛĞæŽÿãAĹãAÛãAęçť'Đ 5,000

æÛĞæŽÿãAñãř;ãAÛãAęãCćãCŌãĈĚãĈijãCũãĈğãĈšãAŦãĈÑãAęãAĐãĈÑãĀĈ

KyotoCorpus

ãAřãřŌæÛëæÛřèAđãAõèÍŸãžÑãCŠãř;èšqãAñãĀã;ćæĚÑçť'ããĈžægÑæÛĞæĈĚããšãAñãAđ'ãAĐãAęãAř 40,000

æÛĞãAñãĀAęĚřèIdéãĚægÑéĀãCžãĚšãRĈçĚğéÛćãĈãAñãAđ'ãAĐãAęãAřãAĹãAõãAĚãAqãAõçť'Đ 10,000

æÛĞãAñãCćãCŌãĈĚãĈijãCũãĈğãĈšãAŦãĈÑãAęãAĐãĈÑãĀĈ

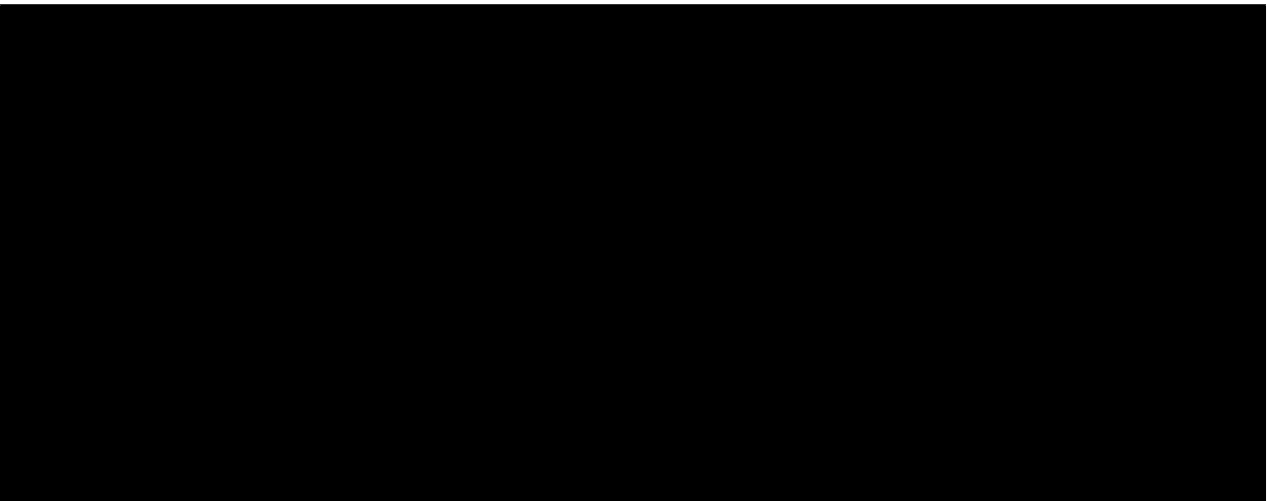
AnnotatedFKCCorpus ãAřãÿĀèĹñãAõãžžãĀĚãAÑãCL'èZEãĈAãCL'ãĈÑãA§ãÿ■æžĀãĈĚãĈ■ãĈZãĈĹçť'Đ 1,300

æÛĞãAñãř;ãAÛãAęãCćãCŌãĈĚãĈijãCũãĈğãĈšãĈSëãÑãAčãA§ãĈšãĈijãĈŘãĈŽãĈZãAęãĈŽãĈãĈÑãĀĈ

ãAĹãAŁãĀAęĚřèIdéãĚægÑéĀãCžãĚšãRĈçĚğéÛćãĈãAõãCćãCŌãĈĚãĈijãCũãĈğãĈšãAř <rel>

ãĈĚãĈřãAñãĈĹãAčãAęëãÑãĈŘãĈÑãAęãAĐãĈÑãĀĈ

KWDLC ãAõã;Ñ:



(continues on next page)

(continued from previous page)



CHAPTER 5

Usage

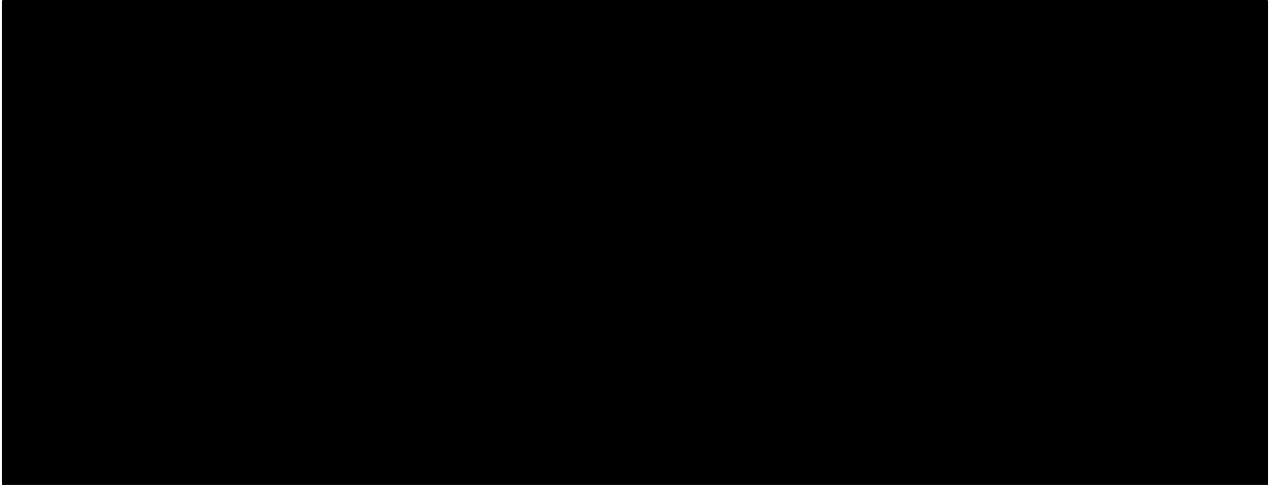
ayLeÍYãAõã¿NãAõãČGãČijaČĚãAŇãĚãAčãAšãČŤãČqãČd'ãČń
ãČŠel■ãAĚë¿ijaČĀããŕ'ãŘĹ

w201106-0000060050.knp

ãČŮãČ■ãČŕãČŕ'ãČããAõãGžãŁŽçŤŘæđIJ

(continues on next page)

(continued from previous page)



```
kyoto āĈšāĈđāĈšāĈĹāĈŠā;ŁċŦīāAŹāĈNāAſāAīāAğāĀAāĈšāĈijāĈŚāĈZāAōāEĖāōZāĈŠēāĴċđ'žāAŮāAſāĈĹāĈšāĈijāĈŚāĈZāĈŠāĹ
```

6.1 Browsing files

- `kyoto show:` `KNP` `āĈŦāCqāĈđ'āĈnāAōāEĖāōZāĈŚāĈĹāĈijā;ċāijRāAğēāĴċđ'ž`
(`āĈĠāĈĈāĈnāĈŕāĈĹāĈĹāĈŠāĖNĠāōZāAŮāAſāāt'āŦĹāĀAāŦnāAĴāĈNāĈNāĖīāAēāAōāĈŦāCqāĈđ'āĈnāĈŠēāĴċđ'ž`)

- `kyoto list:` `āNĠāōZāAŦāĈNāAſāĈĠāĈĈāĈnāĈŕāĈĹāĈĹāAŦāŦnāAĴāĈNāĈNāĖŮĠāZŷ ID āĈŠāĹŮāNŹ`

6.2 Processing Corpus

`āĈšāĈijāĈŚāĈZāĈŠēġċæđRāAŮāĀAēŁ;āĹāāAōċŦ'āæĀğāĈŚāZŸāYŌ` (KNP `āAī JumanDIC āAŦnāŁEēĖA`)

- `kyoto configure:` `āĈšāĈijāĈŚāĈZāAōāĈĠāĈĈāĈnāĈŕāĈĹāĈĹāAŦċŦ'āæĀğāZŸāYŌāAōāAſāĈĹāAō`
`Makefile āĈŚĈŦšāĹŦ`
 - `make` `āĈŠāōšēāNāAŹāĈNāAſāAīāAğāĀAāĈšāĈijāĈŚāĈZāAŦ` `1` `āŮĠāZŷ` `1`
`āĈŦāCqāĈđ'āĈnāAŦāĹēāĹ'sāAŦāĈNāĀA knp/ āĈĠāĈĈāĈnāĈŕāĈĹāĈĹāAŦċŦ'āæĀğāAōāZŸāYŌāAŦāĈNāAſāĈŦāCqāĈđ'`

- `kyoto idsplit:` `āĈšāĈijāĈŚāĈZāĈŚ` `train/dev/test` `āĈŦāCqāĈđ'āĈnāAŦāĹēāĹ's`



6.3 Zsh Completions

```
<virtualenv-path>/share/zsh/site-functions _kyoto_fpath _kyoto_fpath _kyoto_fpath kyoto
_kyoto_fpath _kyoto_fpath _kyoto_fpath (zsh _kyoto_fpath)
```



7.1 kyoto_reader package

7.1.1 Submodules

kyoto_reader.base_phrase module

```

class kyoto_reader.base_phrase.BasePhrase (tag: pyknp.knp.tag.Tag, dmid_offset: int, dtid:
                                             int, sid: str, doc_id: str, parent: Op-
                                             tional[BasePhrase] = None, children: Op-
                                             tional[List[BasePhrase]] = None)

    Bases: object

    æŮĞäÿ■āĀñāĞżçŔĭāAŽāĆNāšžæIJñāŘěāĆŠèąlāAŽāĆřāČřāĆž

    tag
        Tag object in pyknp.
        Type Tag

    sid
        Sentence ID.
        Type str

    dtid
        Document-wide tag ID.
        Type int

    content_dmid
        Document-wide morpheme ID of the content word in the base phrase.
        Type int

    parent
        Dependency parent.

```

Type `Optional[BasePhrase]`

children

Dependency children.

Type `List[BasePhrase]`

__init__ (*tag: pyknp.knp.tag.Tag, dmid_offset: int, dtid: int, sid: str, doc_id: str, parent: Optional[BasePhrase] = None, children: Optional[List[BasePhrase]] = None*)

Parameters

- **tag** (*Tag*) – Tag object in pyknp.
- **dmid_offset** (*int*) – Document-wide morpheme ID of the previous morpheme.
- **dtid** (*int*) – Document-wide tag ID.
- **sid** (*str*) – Sentence ID.
- **doc_id** (*str*) – Document ID.
- **parent** (*Optional[BasePhrase]*) – Dependency parent.
- **children** (*List[BasePhrase]*) – Dependency children.

core

A core expression without ancillary words.

dmid

Document-wide morpheme ID.

dmids

A list of document-wide morpheme IDs.

mrph2dmid

A mapping from morpheme to its document-wide ID.

mrph_list () → `List[pyknp.juman.morpheme.Morpheme]`

A list of morphemes

mrphs

A list of morphemes.

surf

A surface expression.

tid

Tag ID in pyknp.

kyoto_reader.cli module

`kyoto_reader.cli.configure` (*args: argparse.Namespace*)

Create Makefile to preprocess corpus documents.

`kyoto_reader.cli.idspllit` (*args: argparse.Namespace*)

Copy files in a corpus to train, valid (dev), and test directory referring to ID files.

`kyoto_reader.cli.list_` (*args: argparse.Namespace*)

List document IDs which specified path contains.

`kyoto_reader.cli.main` ()

Entry point of CLI commands.

`kyoto_reader.cli.show` (*args: argparse.Namespace*)
 Show the specified document in a tree format.

kyoto_reader.constants module

kyoto_reader.coreference module

class `kyoto_reader.coreference.Entity` (*eid: int, exophor: Optional[str] = None*)

Bases: `object`

A class to represent an entity in coreference. This class manages entity IDs of mentions that refer to this entity.

Parameters

- **eid** (*int*) – An Entity ID.
- **exophor** (*str, optional*) – The kind of exophor if this entity corresponds to some exophor. Otherwise, `None`.

eid

An Entity ID.

Type `int`

exophor

A string to represent exophor, such as `あいつはさういふことをいふ`, `あいつはさういふことをいふ`, and `あいつはさういふことをいふ`.

Type `str, optional`

mentions

A set of mentions that refer to this entity.

Type `Set[Mention]`

mentions_unc

Mentions that have uncertain relation with this entity.

Type `Set[Mention]`

taigen

Whether this entity is `あいつ` or not.

Type `bool, optional`

yougen

Whether this entity is `さう` or not.

Type `bool, optional`

__init__ (*eid: int, exophor: Optional[str] = None*)

Initialize self. See `help(type(self))` for accurate signature.

add_mention (*mention: kyoto_reader.coreference.Mention, uncertain: bool*) → `None`

Add a mention that refers to this entity.

When a non-uncertain mention is added and the mention has already been registered as an uncertain mention, it will be overwritten as non-uncertain.

Parameters

- **mention** (*Mention*) – A mention
- **uncertain** (*bool*) – Whether the mention is uncertain (i.e., annotated with `あいつはさういふことをいふ`).

knp_string
KNP format string of the document.
Type str

doc_id
A document ID.
Type str

cases
Cases to extract.
Type Collection[str]

corefs
Coreference relations to extract.
Type Collection[str]

extract_nes
Whether to extract named entities.
Type bool

sid2sentence
A mapping from a sentence ID to the corresponding sentence.
Type Dict[str, Sentence]

mentions
A mapping from a document-wide tag ID to the corresponding mention.
Type Dict[int, Mention]

entities
A mapping from a entity ID to the corresponding entity.
Type Dict[int, Entity]

named_entities
Extracted named entities.
Type List[NamedEntity]

__init__ (knp_string: str, doc_id: str, cases: Collection[str], corefs: Collection[str], relax_cases: bool, extract_nes: bool, use_pas_tag: bool) → None
Initialize self. See help(type(self)) for accurate signature.

bnst_list () → List[pyknp.knp.bunsetsu.Bunsetsu]
Return list of Bunsetsu object in pyknp.

bp_list () → List[kyoto_reader.base_phrase.BasePhrase]
Return list of base phrases.

draw_tree (sid: Optional[str] = None, coreference: bool = True, fh: Optional[TextIO] = None) → None
Write out the PAS and coreference relations in the specified sentence in a tree format.
If sid is not specified, write out trees in all the sentences in this document.

Parameters

- **sid** (str, optional) – A sentence ID of the target sentence.
- **coreference** (bool) – If True, write out coreference relations as well.

- **fh** (*TextIO*, *optional*) – The output stream.

get_arguments (*predicate*: *kyoto_reader.base_phrase.BasePhrase*, *relax*: *bool* = *False*, *include_optional*: *bool* = *False*) → Dict[str, List[kyoto_reader.pas.BaseArgument]]

Return all the arguments that the given predicate has.

Parameters

- **predicate** (*Predicate*) – A predicate.
- **relax** (*bool*) – If True, return arguments that have a coreference relation with the arguments the predicate has.
- **include_optional** (*bool*) – If True, return adverbial arguments such as あいあいざあらああい as well.

Returns A mapping from a case to arguments.

Return type Dict[str, List[BaseArgument]]

get_entities (*bp*: *kyoto_reader.base_phrase.BasePhrase*, *include_uncertain*: *bool* = *False*) → List[kyoto_reader.coreference.Entity]

Return list of entities that the specified mention refers to. The mention is given as a type of BasePhrase.

Parameters

- **bp** (*BasePhrase*) – A base phrase corresponds to the mention.
- **include_uncertain** (*bool*) – Whether to return entities that has uncertain relation with the mention.

get_predicates () → List[kyoto_reader.base_phrase.BasePhrase]

Return list of predicates.

get_siblings (*mention*: *kyoto_reader.coreference.Mention*, *relax*: *bool* = *False*) → Set[kyoto_reader.coreference.Mention]

Return all the mentions that have coreference chains with the specified mention.

Parameters

- **mention** (*Mention*) – A mention.
- **relax** (*bool*) – If True, return coreferent mentions as well.

Returns A set of mentions.

Return type Set[Mention]

mrph2dmid

A mapping from morpheme to its document-wide ID.

mrph_list () → List[pyknp.juman.morpheme.Morpheme]

Return list of Morpheme object in pyknp.

pas_list () → List[kyoto_reader.pas.Pas]

Return list of predicate-argument structures.

sentences

List of sentences in this document.

Returns List[Sentence]

stat () → dict

Calculate various kinds of statistics of this document.

surf

A surface expression of this document.

tag_list () → List[pyknp.knp.tag.Tag]
Return list of Tag object in pyknp.

kyoto_reader.ne module

class kyoto_reader.ne.**NamedEntity** (*category: str, name: str, sentence: kyoto_reader.sentence.Sentence, mid_range: range, mrph2dmid: Dict[pyknp.juman.morpheme.Morpheme, int]*)

Bases: object

A class to represent a named entity (NE).

Parameters

- **category** (*str*) – A category of a NE.
- **name** (*str*) – A name of a NE.
- **sentence** (*Sentence*) – A sentence that contains a NE.
- **mid_range** (*range*) – A range of IDs of morphemes that constitute a NE.
- **mrph2dmid** (*dict*) – A mapping from morpheme to its document-wide ID.

category

A category of a NE.

Type str

name

A name of a NE.

Type str

sid

A sentence ID of a sentence that contains a NE.

Type str

mid_range

A range of IDs of morphemes that constitute a NE.

Type range

dmid_range

A range of document-wide IDs of morphemes that constitute a NE.

Type range

__init__ (*category: str, name: str, sentence: kyoto_reader.sentence.Sentence, mid_range: range, mrph2dmid: Dict[pyknp.juman.morpheme.Morpheme, int]*)
Initialize self. See help(type(self)) for accurate signature.

kyoto_reader.pas module

class kyoto_reader.pas.**Argument** (*bp: kyoto_reader.base_phrase.BasePhrase, dep_type: str, mode: str*)

Bases: kyoto_reader.base_phrase.BasePhrase, kyoto_reader.pas.BaseArgument

æŮĜäy■ãAñãĜžçŔĭãAžãĈŇ(ãd'ŮçŦŇãAğãAřãAřãAĎ)éãĚãĈŠëáĭãAžãĈĭãĈŮãĈyãĈğãĈřãĈĹ

Parameters

- **bp** (BasePhrase) – âşşæIĴnăŘě
- **dep_type** (*str*) – äŧĆăĈĽăRŮăAşŞăĆŧăĈd’ăĈŮ (âĂIJovertăĂĬ, âĂIJdepăĂĬ, âĂIJintraăĂĬ, âĂIJinterăĂĬ, âĂIJexoăĂĬ)
- **mode** (*str*) – âĈĈăĈĬjăĈĽ

```
__init__(bp: kyoto_reader.base_phrase.BasePhrase, dep_type: str, mode: str) → None
```

Args: tag (Tag): Tag object in pyknp. dmid_offset (int): Document-wide morpheme ID of the previous morpheme. dtid (int): Document-wide tag ID. sid (str): Sentence ID. doc_id (str): Document ID. parent (Optional[BasePhrase]): Dependency parent. children (List[BasePhrase]): Dependency children.

```
class kyoto_reader.pas.BaseArgument (dep_type: str, mode: str)
```

Bases: object

A base class for all kinds of arguments

```
__init__(dep_type: str, mode: str)
```

Initialize self. See help(type(self)) for accurate signature.

is_special

```
class kyoto_reader.pas.Pas (pred_bp: kyoto_reader.base_phrase.BasePhrase)
```

Bases: object

A class to represent a predicate-argument structure (PAS).

Parameters **pred_bp** (BasePhrase) – eřřeldāAłāAłāĆNāšžæIJnāŘě

predicate

è£řèłđ

Type Predicate

arguments

æăijãAíéăĚ

Type Dict[str, List[BaseArgument]]

```
__init__ (pred_bp: kyoto_reader.base_phrase.BasePhrase)
```

Initialize self. See `help(type(self))` for accurate signature.

add_argument (*case: str, bp: kyoto_reader.base_phrase.BasePhrase, mode: str*)
$$\text{add_special_argument} (case: str, exophor: str, eid: int, mode: str) \rightarrow \text{None}$$

dmid

èřřeldãAóäy■ãAóãEĚãóželdã;çæĚŇçťããAóæŮĚãŽvãĈňãĈŽãĈňã;çæĚŇçťãID

dtid

A document-wide tag ID.

```
set_arguments_optional (case: str) → None
```

sid

A sentence ID

```
class kyoto_reader.pas.SpecialArgument (exophor: str, eid: int, mode: str)
```

Bases: *kyoto_reader.pas.BaseArgument*

âd' ŮčTŇăĈŠæŇĠăAZéăĚăĈŠèaĭăAZăĈĭăĈŮăĈvăĈgăĈrăĈĹ

Parameters

- **exophor** (*str*) – áð'ÚçTŇčĚǵâŁIJeł'd (äy■çL'záoŽ:ăzzāAłāAł')
- **eid** (*int*) – áð'ÚçTŇčĚǵâŁIJeł'đāAōāĆíČšāČĚāČćāČĚāČēID

- **mode** (*str*) – `~ČčăČĭjăČĹ`

__init__ (*exophor: str, eid: int, mode: str*)

Initialize self. See help(type(self)) for accurate signature.

kyoto_reader.reader module

class `kyoto_reader.reader.ArchiveHandler` (*path: pathlib.Path*)

Bases: `object`

__init__ (*path: pathlib.Path*) → `None`

Initialize self. See help(type(self)) for accurate signature.

classmethod **is_supported_path** (*path: pathlib.Path*) → `bool`

open () → `Union[tarfile.TarFile, zipfile.ZipFile]`

open_member (*archive: Union[tarfile.TarFile, zipfile.ZipFile], member: str*) → `BinaryIO`

class `kyoto_reader.reader.ArchiveType`

Bases: `enum.Enum`

Enum for file collection types.

TAR_GZ = `'tar.gz'`

ZIP = `'zip'`

class `kyoto_reader.reader.FileHandler` (*path: pathlib.Path*)

Bases: `object`

__init__ (*path: pathlib.Path*) → `None`

Initialize self. See help(type(self)) for accurate signature.

content_basename

open (**args, **kwargs*) → `TextIO`

class `kyoto_reader.reader.FileType`

Bases: `enum.Enum`

Enum for file types.

GZ = `'gz'`

UNCOMPRESSED = `''`

class `kyoto_reader.reader.KyotoReader` (*source: Union[pathlib.Path, str], target_cases: Optional[Collection[str]] = None, target_corefs: Optional[Collection[str]] = None, extract_nes: bool = True, relax_cases: bool = False, use_pas_tag: bool = False, knp_ext: str = '.knp', pickle_ext: str = '.pkl', n_jobs: int = -1, did_from_sid: bool = True*)

Bases: `object`

A class to manage a set of corpus documents. Compressed file is supported. However, nested compression (e.g. .knp.gz in zip file) is not supported.

Parameters

- **source** (*Union[Path, str]*) – `~řĳčšqăAőăŮĞæŽÿăAÿăAőăČŠăČzăĂČăČĞăČčăČňăČřăČĹăČĹăĂŇăŇĞăő`
- **target_cases** (*Optional[Collection[str]]*) – `æŁ;ăĜžăAőăřĳčšqăĂĹăĂŽăČŇăăĭjăĂČ` (default: `ăĚĹăĂęăAőăăĭj`)

- **target_corefs** (*Optional[Collection[str]]*) – æŁĵăĜžăAđăřĹěśăqăĂlăĂZăĆŃăĔśăŘĈĉĖĚăċăĊ(=ăĂłăĂł')ăĂĊ(default: äĤlăAęăAđăĬăċăĊ)
- **extract_nes** (*bool*) – äŻžăIJL'ėăĲĕŖĴăĊŚăĈşăćĳăĈŚăĈżăĂŃăĎĊĻ'æŁĵăĠğžăĂZăĆŃăĂŃăĂł'ăĂĒăĂŃ(default: True)
- **relax_cases** (*bool*) – äĈňăĽ'ŠăăĳăĂłăĂł'ăĈŚăĈňăăĳăĂłăĂŮăĂęăĻ'săĂĒăĂŃ(default: False)
- **knp_ext** (*str*) – KWDLČ äĂĵăĂŞăĂŕ KC äĈŦăĈăăĊď'ăĈňăĂđăŅăăĳăġă■Ŕ (default: knp)
- **pickle_ext** (*str*) – Document äĈŚ pickle äĳăăĳăŔăĂğĖē■ăĈĂăăť'ăŔĻăĂđăŅăăĳăġă■Ŕ (default:.pkl)
- **use_pas_tag** (*bool*) – <rel>ăĈĤăĈŕăĂŃăĎĊĻ'ăĂğăĂŕăĂłăĂŕăăĂĂ<ēĤŕēĥđéăĔăğŃéĂă:>ăĈĤăĈŕăĂŃăĎĊĻ'PAS äĈŚēĭ■ăĈĂăĂŃăĂł'ăĂĒăĂŃ(default: False)
- **n_jobs** (*int*) – æŰĞăŽÿăĈŚēĭ■ăĂĕēĲĵăĈĂăĠğęĤŔĒăĂđăăÿăĹŮăŦŕăĂĈ0:ăÿăĹŮăĠğęĤŔĒăĂłăĂŮăăĂ-1: äĈşăĈċăŦŕ (default: -1)
- **did_from_sid** (*bool*) – æŰĞăŽÿIDăĈŚăŰĞăŽÿăă■ăĂđS-IDăĂŃăĎĊĻ'ăśşăăđŽăĂZăĈŃ (default: True)

[illegible]

```

__init__(source: Union[pathlib.Path, str], target_cases: Optional[Collection[str]] = None, target_corefs: Optional[Collection[str]] = None, extract_nes: bool = True, relax_cases: bool = False, use_pas_tag: bool = False, knp_ext: str = '.knp', pickle_ext: str = '.pkl', n_jobs: int = -1, did_from_sid: bool = True) → None
Initialize self. See help(type(self)) for accurate signature.

```

```
get knp (did: str) → str
```

process_all_documents	(<i>n_jobs</i> :	<i>Optional[int]</i>	=	<i>None</i>)	→
		List[Optional[kyoto_reader.document.Document]]			
Process all documents that KyotoReader has loaded.					

Parameters `n_jobs` (*int*) – The number of processes spawned to finish this task. (default: inherit from self)

```
process_document (doc_id: str, archive: Union[tarfile.TarFile, zipfile.ZipFile, None] = None) →
    Optional[kyoto_reader.document.Document]
    Process one document following the given document ID.
```

Parameters

- **doc_id**(*str*) – An ID of a document to process.
- **archive**(*Optional[ArchiveFile]*) – An archive to read the document from.

```
process_documents (doc_ids: Iterable[str], n_jobs: Optional[int] = None) →  
List[Optional[kyoto_reader.Document]]  
Process multiple documents following the given document IDs.
```

Parameters

- **doc_ids** (*List[str]*) – IDs of documents to process.

- **n_jobs** (*int*) – The number of processes spawned to finish this task. (default: inherit from self)

kyoto_reader.sentence module

class `kyoto_reader.sentence.Sentence` (*knp_string: str, dtid_offset: int, dmid_offset: int, doc_id: str*)

Bases: `object`

A class to represent a single sentence.

blist

BList object of pyknp.

Type `BList`

doc_id

The document ID of this sentence.

Type `str`

bps

Base phrases in this sentence.

Type `List[BasePhrase]`

__init__ (*knp_string: str, dtid_offset: int, dmid_offset: int, doc_id: str*) → `None`

Parameters

- **knp_string** (*str*) – KNP format string of this sentence.
- **dtid_offset** (*int*) – The document-wide tag ID of the previous base phrase.
- **dmid_offset** (*int*) – The document-wide morpheme ID of the previous morpheme.
- **doc_id** (*str*) – The document ID of this sentence.

bnst_list ()

Return list of Bunsetsu object in pyknp.

dtids

A document-wide tag ID.

mrph2dmid

A mapping from morpheme to its document-wide ID.

mrph_list ()

Return list of Morpheme object in pyknp.

sid

A sentence ID.

surf

A surface expression

tag_list ()

Return list of Tag object in pyknp.

7.1.2 Module contents

CHAPTER 8

Author/Contact

äžňěČĭăd'gā■ē ézŠæł'ŇăČzèd'ŽăČzæĬSèĎĜčăTçł'űăăd' (contact at nlp.ist.i.kyoto-u.ac.jp)

- Nobuhiro Ueda

CHAPTER 9

Indices and Tables

- `genindex`
- `modindex`
- `search`

k

- `kyoto_reader`, 23
- `kyoto_reader.base_phrase`, 13
- `kyoto_reader.cli`, 14
- `kyoto_reader.constants`, 15
- `kyoto_reader.coreference`, 15
- `kyoto_reader.document`, 16
- `kyoto_reader.ne`, 19
- `kyoto_reader.pas`, 19
- `kyoto_reader.reader`, 21
- `kyoto_reader.sentence`, 23

Symbols

`__init__()` (*kyoto_reader.base_phrase.BasePhrase* method), 14
`__init__()` (*kyoto_reader.coreference.Entity* method), 15
`__init__()` (*kyoto_reader.coreference.Mention* method), 16
`__init__()` (*kyoto_reader.document.Document* method), 17
`__init__()` (*kyoto_reader.ne.NamedEntity* method), 19
`__init__()` (*kyoto_reader.pas.Argument* method), 20
`__init__()` (*kyoto_reader.pas.BaseArgument* method), 20
`__init__()` (*kyoto_reader.pas.Pas* method), 20
`__init__()` (*kyoto_reader.pas.SpecialArgument* method), 21
`__init__()` (*kyoto_reader.reader.ArchiveHandler* method), 21
`__init__()` (*kyoto_reader.reader.FileHandler* method), 21
`__init__()` (*kyoto_reader.reader.KyotoReader* method), 22
`__init__()` (*kyoto_reader.sentence.Sentence* method), 23

A

`add_argument()` (*kyoto_reader.pas.Pas* method), 20
`add_mention()` (*kyoto_reader.coreference.Entity* method), 15
`add_special_argument()` (*kyoto_reader.pas.Pas* method), 20
`all_eids` (*kyoto_reader.coreference.Mention* attribute), 16
`all_mentions` (*kyoto_reader.coreference.Entity* attribute), 15
`ArchiveHandler` (class in *kyoto_reader.reader*), 21
`ArchiveType` (class in *kyoto_reader.reader*), 21
`Argument` (class in *kyoto_reader.pas*), 19

`arguments` (*kyoto_reader.pas.Pas* attribute), 20

B

`BaseArgument` (class in *kyoto_reader.pas*), 20
`BasePhrase` (class in *kyoto_reader.base_phrase*), 13
`blist` (*kyoto_reader.sentence.Sentence* attribute), 23
`bnst_list()` (*kyoto_reader.document.Document* method), 17
`bnst_list()` (*kyoto_reader.sentence.Sentence* method), 23
`bp_list()` (*kyoto_reader.document.Document* method), 17
`bps` (*kyoto_reader.sentence.Sentence* attribute), 23

C

`cases` (*kyoto_reader.document.Document* attribute), 17
`category` (*kyoto_reader.ne.NamedEntity* attribute), 19
`children` (*kyoto_reader.base_phrase.BasePhrase* attribute), 14
`configure()` (in module *kyoto_reader.cli*), 14
`content_basename` (*kyoto_reader.reader.FileHandler* attribute), 21
`content_dmid` (*kyoto_reader.base_phrase.BasePhrase* attribute), 13
`core` (*kyoto_reader.base_phrase.BasePhrase* attribute), 14
`corefs` (*kyoto_reader.document.Document* attribute), 17

D

`dmid` (*kyoto_reader.base_phrase.BasePhrase* attribute), 14
`dmid` (*kyoto_reader.pas.Pas* attribute), 20
`dmid_range` (*kyoto_reader.ne.NamedEntity* attribute), 19
`dmids` (*kyoto_reader.base_phrase.BasePhrase* attribute), 14
`doc_id` (*kyoto_reader.document.Document* attribute), 17

`doc_id` (*kyoto_reader.sentence.Sentence* attribute), 23
`Document` (class in *kyoto_reader.document*), 16
`draw_tree()` (*kyoto_reader.document.Document* method), 17
`dtid` (*kyoto_reader.base_phrase.BasePhrase* attribute), 13
`dtid` (*kyoto_reader.pas.Pas* attribute), 20
`dtids` (*kyoto_reader.sentence.Sentence* attribute), 23

E

`eid` (*kyoto_reader.coreference.Entity* attribute), 15
`eids` (*kyoto_reader.coreference.Mention* attribute), 16
`eids_unc` (*kyoto_reader.coreference.Mention* attribute), 16
`entities` (*kyoto_reader.document.Document* attribute), 17
`Entity` (class in *kyoto_reader.coreference*), 15
`exophor` (*kyoto_reader.coreference.Entity* attribute), 15
`extract_nes` (*kyoto_reader.document.Document* attribute), 17

F

`FileHandler` (class in *kyoto_reader.reader*), 21
`FileType` (class in *kyoto_reader.reader*), 21

G

`get_arguments()` (*kyoto_reader.document.Document* method), 18
`get_entities()` (*kyoto_reader.document.Document* method), 18
`get_knp()` (*kyoto_reader.reader.KyotoReader* method), 22
`get_predicates()` (*kyoto_reader.document.Document* method), 18
`get_siblings()` (*kyoto_reader.document.Document* method), 18
`GZ` (*kyoto_reader.reader.FileType* attribute), 21

I

`idsplit()` (in module *kyoto_reader.cli*), 14
`is_special` (*kyoto_reader.coreference.Entity* attribute), 16
`is_special` (*kyoto_reader.pas.BaseArgument* attribute), 20
`is_supported_path()` (*kyoto_reader.reader.ArchiveHandler* class method), 21
`is_uncertain_to()` (*kyoto_reader.coreference.Mention* method), 16

K

`knp_string` (*kyoto_reader.document.Document* attribute), 16
`kyoto_reader` (module), 23
`kyoto_reader.base_phrase` (module), 13
`kyoto_reader.cli` (module), 14
`kyoto_reader.constants` (module), 15
`kyoto_reader.coreference` (module), 15
`kyoto_reader.document` (module), 16
`kyoto_reader.ne` (module), 19
`kyoto_reader.pas` (module), 19
`kyoto_reader.reader` (module), 21
`kyoto_reader.sentence` (module), 23
`KyotoReader` (class in *kyoto_reader.reader*), 21

L

`list_()` (in module *kyoto_reader.cli*), 14

M

`main()` (in module *kyoto_reader.cli*), 14
`Mention` (class in *kyoto_reader.coreference*), 16
`mentions` (*kyoto_reader.coreference.Entity* attribute), 15
`mentions` (*kyoto_reader.document.Document* attribute), 17
`mentions_unc` (*kyoto_reader.coreference.Entity* attribute), 15
`mid_range` (*kyoto_reader.ne.NamedEntity* attribute), 19
`mrph2dmid` (*kyoto_reader.base_phrase.BasePhrase* attribute), 14
`mrph2dmid` (*kyoto_reader.document.Document* attribute), 18
`mrph2dmid` (*kyoto_reader.sentence.Sentence* attribute), 23
`mrph_list()` (*kyoto_reader.base_phrase.BasePhrase* method), 14
`mrph_list()` (*kyoto_reader.document.Document* method), 18
`mrph_list()` (*kyoto_reader.sentence.Sentence* method), 23
`mrphs` (*kyoto_reader.base_phrase.BasePhrase* attribute), 14

N

`name` (*kyoto_reader.ne.NamedEntity* attribute), 19
`named_entities` (*kyoto_reader.document.Document* attribute), 17
`NamedEntity` (class in *kyoto_reader.ne*), 19

O

`open()` (*kyoto_reader.reader.ArchiveHandler* method), 21

`open()` (*kyoto_reader.reader.FileHandler method*), 21
`open_member()` (*kyoto_reader.reader.ArchiveHandler method*), 21

P

`parent` (*kyoto_reader.base_phrase.BasePhrase attribute*), 13
Pas (*class in kyoto_reader.pas*), 20
`pas_list()` (*kyoto_reader.document.Document method*), 18
`predicate` (*kyoto_reader.pas.Pas attribute*), 20
`process_all_documents()` (*kyoto_reader.reader.KyotoReader method*), 22
`process_document()` (*kyoto_reader.reader.KyotoReader method*), 22
`process_documents()` (*kyoto_reader.reader.KyotoReader method*), 22

R

`remove_mention()` (*kyoto_reader.coreference.Entity method*), 16

S

Sentence (*class in kyoto_reader.sentence*), 23
`sentences` (*kyoto_reader.document.Document attribute*), 18
`set_arguments_optional()` (*kyoto_reader.pas.Pas method*), 20
`show()` (*in module kyoto_reader.cli*), 14
`sid` (*kyoto_reader.base_phrase.BasePhrase attribute*), 13
`sid` (*kyoto_reader.ne.NamedEntity attribute*), 19
`sid` (*kyoto_reader.pas.Pas attribute*), 20
`sid` (*kyoto_reader.sentence.Sentence attribute*), 23
`sid2sentence` (*kyoto_reader.document.Document attribute*), 17
SpecialArgument (*class in kyoto_reader.pas*), 20
`stat()` (*kyoto_reader.document.Document method*), 18
`surf` (*kyoto_reader.base_phrase.BasePhrase attribute*), 14
`surf` (*kyoto_reader.document.Document attribute*), 18
`surf` (*kyoto_reader.sentence.Sentence attribute*), 23

T

`tag` (*kyoto_reader.base_phrase.BasePhrase attribute*), 13
`tag_list()` (*kyoto_reader.document.Document method*), 18
`tag_list()` (*kyoto_reader.sentence.Sentence method*), 23
`taigen` (*kyoto_reader.coreference.Entity attribute*), 15

`TAR_GZ` (*kyoto_reader.reader.ArchiveType attribute*), 21
`tid` (*kyoto_reader.base_phrase.BasePhrase attribute*), 14

U

`UNCOMPRESSED` (*kyoto_reader.reader.FileType attribute*), 21

Y

`yougen` (*kyoto_reader.coreference.Entity attribute*), 15

Z

`ZIP` (*kyoto_reader.reader.ArchiveType attribute*), 21