
pyknp-eventgraph

Kurohashi-Kawahara Lab, Kyoto University

Mar 30, 2021

REFERENCES

1	About	1
2	Requirements	3
3	Installation	5
3.1	API Reference	5
4	Author/Contact	19
4.1	License	19
5	Indices and tables	21
	Python Module Index	23
	Index	25

ABOUT

EventGraph is a development platform for high-level NLP applications in Japanese. The core concept of EventGraph is event, a language information unit that is closely related to predicate-argument structure but more application-oriented. Events are linked to each other based on their syntactic and semantic relations.

REQUIREMENTS

- Python 3.6 or later
- `pyknp`
- `graphviz`

INSTALLATION

To install `pyknp-eventgraph`, use `pip`.

```
$ pip install pyknp-eventgraph
```

or

```
$ git clone https://github.com/ku-nlp/pyknp-eventgraph.git
$ cd pyknp-eventgraph
$ python setup.py install [--prefix=path]
```

3.1 API Reference

3.1.1 `pyknp_eventgraph.eventgraph`

class `pyknp_eventgraph.eventgraph.EventGraph`

Bases: `pyknp_eventgraph.component.Component`

`EventGraph` provides a high-level interface that facilitates NLP application development. The core concept of `EventGraph` is event, a language information unit that is closely related to predicate-argument structure but more application-oriented. Events are linked to each other based on their syntactic and semantic relations.

document: `Document`

A document on which this `EventGraph` is built.

classmethod `build(blist)`

Build an `EventGraph` from language analysis by KNP.

Parameters `blist` (`List[BList]`) – A list of bunsetsu lists, each of which is a result of analysis performed by KNP on a sentence.

Example:

```
from pyknp import KNP
from pyknp_eventgraph import EventGraph

# Parse a document.
document = ['', '']
knp = KNP()
blists = [knp.parse(sentence) for sentence in document]

# Build an EventGraph.
evg = EventGraph.build(blists)
```

Return type *EventGraph*

classmethod `load(f, binary=False)`

Deserialize an EventGraph.

Parameters

- **f** (`Union[TextIO, BinaryIO]`) – A file descriptor.
- **binary** (`bool`) – If true, deserialize an EventGraph using Python’s pickle utility. Otherwise, deserialize an EventGraph using Python’s json utility.

Example:

```
from pyknp_eventgraph import EventGraph

# Load an EventGraph serialized in a JSON format.
with open('evg.json', 'r') as f:
    evg = EventGraph.load(f, binary=False)

# Load an EventGraph serialized by Python's pickle utility.
with open('evg.pkl', 'rb') as f:
    evg = EventGraph.load(f, binary=True)
```

Caution: EventGraph deserialized from a JSON file loses several functionality. To keep full functionality, use Python’s pickle utility for serialization.

Return type *EventGraph*

save (`path, binary=False`)

Save this EventGraph.

Parameters

- **path** (`str`) – An output file path.
- **binary** (`bool`) – If true, serialize this EventGraph using Python’s pickle utility. Otherwise, serialize this EventGraph using Python’s json utility.

Caution: EventGraph deserialized from a JSON file loses several functionality. To keep full functionality, use Python’s pickle utility for serialization.

Return type `None`

to_dict ()

Convert this object into a dictionary.

Return type `dict`

to_string ()

Convert this object into a string.

Return type `str`

property `events`

A list of events.

Return type `List[Event]`

property relations

A list of relations.

Return type `List[Relation]`

property sentences

A list of sentences.

Return type `List[Sentence]`

3.1.2 pyknp_eventgraph.document

class `pyknp_eventgraph.document.Document` (*evg*)
 Bases: `pyknp_eventgraph.component.Component`

A document is a collection of sentences.

evg: EventGraph

An EventGraph built on this document.

sentences: List[Sentence]

A list of sentences in this document.

to_dict()

Convert this object into a dictionary.

Return type `dict`

to_string()

Convert this object into a string.

Return type `str`

3.1.3 pyknp_eventgraph.sentence

class `pyknp_eventgraph.sentence.Sentence` (*document, sid, ssid, blist=None*)
 Bases: `pyknp_eventgraph.component.Component`

A sentence is a collection of events.

document: Document

A document that includes this sentence.

sid: str

An original sentence ID.

ssid: int

A serial sentence ID.

blist: :class:`pyknp.knp.blist.BList`, optional

A list of bunsetsu-s.

events: List[Event]

A list of events in this sentence.

to_dict()

Convert this object into a dictionary.

Return type `dict`

to_string()
Convert this object into a string.
Return type `str`

property mrphs
A tokenized surface string.
Return type `str`

property reps
A representative string.
Return type `str`

property surf
A surface string.
Return type `str`

3.1.4 pyknp_eventgraph.event

class `pyknp_eventgraph.event.Event` (*sentence*, *evid*, *sid*, *ssid*, *start=None*, *head=None*,
end=None)
Bases: `pyknp_eventgraph.component.Component`

Event is the basic information unit of EventGraph. Event is closely related to PAS but more application-oriented with respect to the following points:

- Semantic heaviness: Some predicates are too semantically light for applications to treat as information units. EventGraph constrains an event to have a semantically heavy predicate.
- Rich linguistic features: Linguistic features such as tense and modality are assigned to events.

sentence: `:class:`.Sentence``
A sentence to which this event belongs.

evid: `int`
A serial event ID.

sid: `str`
An original sentence ID.

ssid: `int`
A serial sentence ID.

start: `:class:`pyknp.knp.tag.Tag`, optional`
A start tag.

head: `:class:`pyknp.knp.tag.Tag`, optional`
A head tag.

end: `:class:`pyknp.knp.tag.Tag`, optional`
An end tag.

pas: `PAS, optional`
A predicate argument structure.

outgoing_relations: `List[Relation]`
A list of relations where this event is the modifier.

incoming_relations: `List[Relation]`
A list of relations where this event is the head.

features: **Features**, optional

Linguistic features.

parent: **Event**, optional

A parent event.

children: **List[Event]**

A list of child events.

head_base_phrase: **Token**, optional

A head basic phrase.

content_rep_list_()

A list of content words.

Return type `List[str]`

mrphs_ (*include_modifiers=False*)

A tokenized surface string.

Parameters **include_modifiers** (`bool`) – If true, tokens of events that modify this event will be included.

Return type `str`

mrphs_with_mark_ (*include_modifiers=False*)

A tokenized surface string with marks.

Parameters **include_modifiers** (`bool`) – If true, tokens of events that modify this event will be included.

Return type `str`

normalized_mrphs_ (*include_modifiers=False*)

A tokenized/normalized surface string.

Parameters **include_modifiers** (`bool`) – If true, tokens of events that modify this event will be included.

Return type `str`

normalized_mrphs_with_mark_ (*include_modifiers=False*)

A tokenized/normalized surface string with marks.

Parameters **include_modifiers** (`bool`) – If true, tokens of events that modify this event will be included.

Return type `str`

normalized_mrphs_with_mark_without_exophora_ (*include_modifiers=False*)

A tokenized/normalized surface string with marks but without exophora.

Parameters **include_modifiers** (`bool`) – If true, tokens of events that modify this event will be included.

Return type `str`

normalized_mrphs_without_exophora_ (*include_modifiers=False*)

A tokenized/normalized surface string without exophora.

Parameters **include_modifiers** (`bool`) – If true, tokens of events that modify this event will be included.

Return type `str`

normalized_reps_ (*include_modifiers=False*)

A normalized representative string.

Parameters **include_modifiers** (*bool*) – If true, tokens of events that modify this event will be included.

Return type *str*

normalized_reps_with_mark_ (*include_modifiers=False*)

A normalized representative string with marks.

Parameters **include_modifiers** (*bool*) – If true, tokens of events that modify this event will be included.

Return type *str*

reps_ (*include_modifiers=False*)

A representative string.

Parameters **include_modifiers** (*bool*) – If true, tokens of events that modify this event will be included.

Return type *str*

reps_with_mark_ (*include_modifiers=False*)

A representative string with marks.

Parameters **include_modifiers** (*bool*) – If true, tokens of events that modify this event will be included.

Return type *str*

surf_ (*include_modifiers=False*)

A surface string.

Parameters **include_modifiers** (*bool*) – If true, tokens of events that modify this event will be included.

Return type *str*

surf_with_mark_ (*include_modifiers=False*)

A surface string with marks.

Parameters **include_modifiers** (*bool*) – If true, tokens of events that modify this event will be included.

Return type *str*

to_dict ()

Convert this object into a dictionary.

Return type *dict*

to_string ()

Convert this object into a string.

Return type *str*

property content_rep_list

A list of content words.

Return type *List[str]*

property event_id

An alias to evid.

Return type `int`

property `mrphs`

A tokenized surface string.

Return type `str`

property `mrphs_with_mark`

A tokenized surface string with marks.

Return type `str`

property `normalized_mrphs`

A tokenized/normalized surface string.

Return type `str`

property `normalized_mrphs_with_mark`

A tokenized/normalized surface string with marks.

Return type `str`

property `normalized_mrphs_with_mark_without_exophora`

A tokenized/normalized surface string with marks but without exophora.

Return type `str`

property `normalized_mrphs_without_exophora`

A tokenized/normalized surface string without exophora.

Return type `str`

property `normalized_reps`

A normalized representative string.

Return type `str`

property `normalized_reps_with_mark`

A normalized representative string with marks.

Return type `str`

property `reps`

A representative string.

Return type `str`

property `reps_with_mark`

A representative string with marks.

Return type `str`

property `surf`

A surface string.

Return type `str`

property `surf_with_mark`

A surface string with marks.

Return type `str`

3.1.5 pyknp_eventgraph.pas

```
class pyknp_eventgraph.pas.PAS (event, pas=None)
    Bases: pyknp_eventgraph.component.Component

    A PAS is the core of an event.

    event: Event
        An event that this PAS belongs.

    sid: str
        An original sentence ID.

    ssid: int
        A serial sentence ID.

    pas: :class:`pyknp.knp.pas.Pas`, optional
        A PAS object in pyknp.

    predicate: Predicate
        A predicate.

    arguments: Dict[str, List[Argument]]
        A mapping of a case to arguments.

    to_dict()
        Convert this object into a dictionary.

        Return type dict

    to_string()
        Convert this object into a string.

        Return type str
```

3.1.6 pyknp_eventgraph.predicate

```
class pyknp_eventgraph.predicate.Predicate (pas, type_, head=None)
    Bases: pyknp_eventgraph.component.Component

    A predicate is the core of aPAS.

    pas: PAS
        A PAS that this predicate belongs.

    head: :class:`pyknp.knp.tag.Tag`
        A head tag.

    type_: str
        A type of this predicate.

    head_base_phrase: Token, optional
        A head basic phrase.

    to_dict()
        Convert this object into a dictionary.

        Return type dict

    to_string()
        Convert this object into a string.

        Return type str
```


property adnominal_event_ids

A list of IDs of events modifying this predicate (adnominal).

Return type `List[int]`

property children

A list of child words.

Return type `List[dict]`

property mrphs

A tokenized string.

Return type `str`

property normalized_mrphs

A tokenized/normalized surface string.

Return type `str`

property normalized_reps

A normalized representative string.

Return type `str`

property normalized_surf

A normalized surface string.

Return type `str`

property reps

A representative string.

Return type `str`

property sentential_complement_event_ids

A list of IDs of events modifying this predicate (sentential complement).

Return type `List[int]`

property standard_reps

A standard representative string.

Return type `str`

property surf

A surface string.

Return type `str`

property tag

The tag of the head base phrase.

Return type `Optional[Tag]`

property type

The type of this predicate.

Return type `str`

3.1.7 pyknp_eventgraph.argument

class pyknp_eventgraph.argument.**Argument** (*pas, case, eid, flag, sdist, arg=None*)

Bases: *pyknp_eventgraph.component.Component*

An argument supplements its predicate's information.

pas: **PAS**

A PAS that this argument belongs.

case: **str**

A case.

eid: **int**

An entity ID.

flag: **str**

A flag.

sdist: **int**

The sentence distance between this argument and the predicate.

arg: **:class:`pyknp.knp.pas.Argument`, optional**

An Argument object in pyknp.

head_base_phrase: **Token, optional**

A head basic phrase.

to_dict()

Convert this object into a dictionary.

Return type **dict**

to_string()

Convert this object into a string.

Return type **str**

property adnominal_event_ids

A list of IDs of events modifying this predicate (adnominal).

Return type **List[int]**

property children

A list of child words.

Return type **List[dict]**

property head_reps

A head representative string.

Return type **str**

property mrphs

A tokenized surface string.

Return type **str**

property normalized_mrphs

A tokenized/normalized surface string.

Return type **str**

property normalized_reps

A normalized representative string.

Return type `str`

property normalized_surf
A normalized surface string.

Return type `str`

property reps
A representative string.

Return type `str`

property sentential_complement_event_ids
A list of IDs of events modifying this predicate (sentential complement).

Return type `List[int]`

property surf
A surface string.

Return type `str`

property tag
The tag of the head base phrase.

Return type `Optional[Tag]`

3.1.8 pyknp_eventgraph.features

class `pyknp_eventgraph.features.Features` (*event, modality, tense, negation, state, complement, level=None*)

Bases: `pyknp_eventgraph.component.Component`

Features provides linguistic information of an event.

event: `Event`

An event.

modality: `List[str]`

A list of modality, a linguistic expression that indicates how a write judges and feels about content. Each of item can take either “(volition),” “(invitation),” “(imperative),” “(prohibition),” “: (evaluation: weak),” “: (evaluation: strong),” “- (certainty-subjective),” “- (certainty-epistemic),” “- (certainty-evidential),” “(request-A),” “(request-B),” and “(supposition/hearsay).”

tense: `str`

The place of an event in a time frame, which can take either “(past)” or “(non-past).”

negation: `bool`

If true, this event uses a negative construction.

state: `str`

A type of a predicate, which can take either “(action)” or “(state).”

complement: `bool`

If true, this event modifies an event as a sentential complementizer.

level: `str, optional`

The semantic heaviness of a predicate.

to_dict ()

Convert this object into a dictionary.

Return type `dict`

to_string()
Convert this object into a string.
Return type `str`

3.1.9 pyknp_eventgraph.relation

class `pyknp_eventgraph.relation.Relation` (*modifier, head, label, surf, head_tid, reliable*)
Bases: `pyknp_eventgraph.component.Component`

A relation connects two events. Relations fall into two major divisions: syntactic and discourse relations. Syntactic relations can be used by application developers to, for example, construct a larger information unit by merging a modifier event to the modifiee, while discourse relations offer more pragmatic information, paving the way for deep language understanding.

modifier: `Event`
A modifier event.

head: `Event`
A head event.

label: `str`
A relation label. Syntactic relation labels include “(adnominal relation,” “(sentential complement,” “(parallel),” and “(dependency).” On the other hand, discourse relation labels include “(cause/reason,” “(purpose,” “(condition,” “(ground,” “(contrast,” and “(concession).”

surf: `str`
A surface string.

head_tid: `int`
A tag ID.

reliable: `bool`
If true, a syntactic dependency is not ambiguous.

to_dict()
Convert this object into a dictionary.
Return type `dict`

to_string()
Convert this object into a string.
Return type `str`

3.1.10 pyknp_eventgraph.component

class `pyknp_eventgraph.component.Component`
Bases: `abc.ABC`

The base of EventGraph components.

abstract to_dict()
Convert this object into a dictionary.
Return type `dict`

abstract to_string()
Convert this object into a string.
Return type `str`

3.1.11 pyknp_eventgraph.utils

`pyknp_eventgraph.utils.read_knp_result_file(filename)`

Read a KNP result file.

Parameters `filename` (*str*) – A filename.

Return type `List[BList]`

Returns A list of `pyknp.knp.blist.BList` objects.

3.1.12 pyknp_eventgraph.visualizer

`pyknp_eventgraph.visualizer.make_image(ev, output, with_detail=True, with_original_text=True)`

Visualize an EventGraph.

Parameters

- **evg** (*EventGraph*) – An EventGraph.
- **output** (*str*) – Path to an output file. The file extension must be ‘.svg’.
- **with_detail** (*bool*) – If true, detail information will be included.
- **with_original_text** (*bool*) – If true, original sentences will be included.

AUTHOR/CONTACT

Kurohashi-Kawahara Lab, Kyoto University (contact@nlp.ist.i.kyoto-u.ac.jp)

- Hirokazu Kiyomaru

4.1 License

BSD 3-Clause License

Copyright (c) 2019, Kyoto University All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- Neither the name of the copyright holder nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS “AS IS” AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

INDICES AND TABLES

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

PYTHON MODULE INDEX

p

`pyknp-eventgraph`, [5](#)
`pyknp_eventgraph.utils`, [17](#)
`pyknp_eventgraph.visualizer`, [17](#)

A

adnominal_event_ids()
 (*pyknp_eventgraph.argument.Argument* property), 14
 adnominal_event_ids()
 (*pyknp_eventgraph.predicate.Predicate* property), 12
 arg (*pyknp_eventgraph.argument.Argument* attribute), 14
 Argument (*class in pyknp_eventgraph.argument*), 14
 arguments (*pyknp_eventgraph.pas.PAS* attribute), 12

B

blist (*pyknp_eventgraph.sentence.Sentence* attribute), 7
 build() (*pyknp_eventgraph.eventgraph.EventGraph* class method), 5

C

case (*pyknp_eventgraph.argument.Argument* attribute), 14
 children (*pyknp_eventgraph.event.Event* attribute), 9
 children() (*pyknp_eventgraph.argument.Argument* property), 14
 children() (*pyknp_eventgraph.predicate.Predicate* property), 13
 complement (*pyknp_eventgraph.features.Features* attribute), 15
 Component (*class in pyknp_eventgraph.component*), 16
 content_rep_list()
 (*pyknp_eventgraph.event.Event* property), 10
 content_rep_list()
 (*pyknp_eventgraph.event.Event* method), 9

D

Document (*class in pyknp_eventgraph.document*), 7
 document (*pyknp_eventgraph.eventgraph.EventGraph* attribute), 5
 document (*pyknp_eventgraph.sentence.Sentence* attribute), 7

E

eid (*pyknp_eventgraph.argument.Argument* attribute), 14
 end (*pyknp_eventgraph.event.Event* attribute), 8
 Event (*class in pyknp_eventgraph.event*), 8
 event (*pyknp_eventgraph.features.Features* attribute), 15
 event (*pyknp_eventgraph.pas.PAS* attribute), 12
 event_id() (*pyknp_eventgraph.event.Event* property), 10
 EventGraph (*class in pyknp_eventgraph.eventgraph*), 5
 events (*pyknp_eventgraph.sentence.Sentence* attribute), 7
 events() (*pyknp_eventgraph.eventgraph.EventGraph* property), 6
 evg (*pyknp_eventgraph.document.Document* attribute), 7
 evid (*pyknp_eventgraph.event.Event* attribute), 8

F

Features (*class in pyknp_eventgraph.features*), 15
 features (*pyknp_eventgraph.event.Event* attribute), 8
 flag (*pyknp_eventgraph.argument.Argument* attribute), 14

H

head (*pyknp_eventgraph.event.Event* attribute), 8
 head (*pyknp_eventgraph.predicate.Predicate* attribute), 12
 head (*pyknp_eventgraph.relation.Relation* attribute), 16
 head_base_phrase (*pyknp_eventgraph.argument.Argument* attribute), 14
 head_base_phrase (*pyknp_eventgraph.event.Event* attribute), 9
 head_base_phrase (*pyknp_eventgraph.predicate.Predicate* attribute), 12
 head_reps() (*pyknp_eventgraph.argument.Argument* property), 14
 head_tid (*pyknp_eventgraph.relation.Relation* attribute), 16

I

incoming_relations
(pyknp_eventgraph.event.Event attribute),
8

L

label (pyknp_eventgraph.relation.Relation attribute),
16
level (pyknp_eventgraph.features.Features attribute),
15
load() (pyknp_eventgraph.eventgraph.EventGraph
class method), 6

M

make_image() (in module
pyknp_eventgraph.visualizer), 17
modality (pyknp_eventgraph.features.Features at-
tribute), 15
modifier (pyknp_eventgraph.relation.Relation at-
tribute), 16
module
pyknp_eventgraph.utils, 17
pyknp_eventgraph.visualizer, 17
pyknp-eventgraph, 5
mrphs() (pyknp_eventgraph.argument.Argument prop-
erty), 14
mrphs() (pyknp_eventgraph.event.Event property), 11
mrphs() (pyknp_eventgraph.predicate.Predicate prop-
erty), 13
mrphs() (pyknp_eventgraph.sentence.Sentence prop-
erty), 8
mrphs_() (pyknp_eventgraph.event.Event method), 9
mrphs_with_mark() (pyknp_eventgraph.event.Event
property), 11
mrphs_with_mark_()
(pyknp_eventgraph.event.Event method),
9

N

negation (pyknp_eventgraph.features.Features at-
tribute), 15
normalized_mrphs()
(pyknp_eventgraph.argument.Argument prop-
erty), 14
normalized_mrphs()
(pyknp_eventgraph.event.Event property),
11
normalized_mrphs()
(pyknp_eventgraph.predicate.Predicate prop-
erty), 13
normalized_mrphs_()
(pyknp_eventgraph.event.Event method),
9

normalized_mrphs_with_mark()
(pyknp_eventgraph.event.Event property),
11
normalized_mrphs_with_mark_()
(pyknp_eventgraph.event.Event method),
9
normalized_mrphs_with_mark_without_exophora()
(pyknp_eventgraph.event.Event property), 11
normalized_mrphs_with_mark_without_exophora_()
(pyknp_eventgraph.event.Event method), 9
normalized_mrphs_without_exophora()
(pyknp_eventgraph.event.Event property), 11
normalized_mrphs_without_exophora_()
(pyknp_eventgraph.event.Event method), 9
normalized_reps()
(pyknp_eventgraph.argument.Argument prop-
erty), 14
normalized_reps() (pyknp_eventgraph.event.Event
property), 11
normalized_reps()
(pyknp_eventgraph.predicate.Predicate prop-
erty), 13
normalized_reps_()
(pyknp_eventgraph.event.Event method),
9
normalized_reps_with_mark()
(pyknp_eventgraph.event.Event property),
11
normalized_reps_with_mark_()
(pyknp_eventgraph.event.Event method),
10
normalized_surf()
(pyknp_eventgraph.argument.Argument prop-
erty), 15
normalized_surf()
(pyknp_eventgraph.predicate.Predicate prop-
erty), 13

O

outgoing_relations
(pyknp_eventgraph.event.Event attribute),
8

P

parent (pyknp_eventgraph.event.Event attribute), 9
PAS (class in pyknp_eventgraph.pas), 12
pas (pyknp_eventgraph.argument.Argument attribute),
14
pas (pyknp_eventgraph.event.Event attribute), 8
pas (pyknp_eventgraph.pas.PAS attribute), 12
pas (pyknp_eventgraph.predicate.Predicate attribute),
12
Predicate (class in pyknp_eventgraph.predicate), 12
predicate (pyknp_eventgraph.pas.PAS attribute), 12

pyknp_eventgraph.utils
module, 17

pyknp_eventgraph.visualizer
module, 17

pyknp-eventgraph
module, 5

R

read_knp_result_file() (in module
pyknp_eventgraph.utils), 17

Relation (class in pyknp_eventgraph.relation), 16

relations() (pyknp_eventgraph.eventgraph.EventGraph
property), 7

reliable (pyknp_eventgraph.relation.Relation at-
tribute), 16

reps() (pyknp_eventgraph.argument.Argument prop-
erty), 15

reps() (pyknp_eventgraph.event.Event property), 11

reps() (pyknp_eventgraph.predicate.Predicate prop-
erty), 13

reps() (pyknp_eventgraph.sentence.Sentence prop-
erty), 8

reps_() (pyknp_eventgraph.event.Event method), 10

reps_with_mark() (pyknp_eventgraph.event.Event
property), 11

reps_with_mark_() (pyknp_eventgraph.event.Event
method), 10

S

save() (pyknp_eventgraph.eventgraph.EventGraph
method), 6

sdist (pyknp_eventgraph.argument.Argument at-
tribute), 14

Sentence (class in pyknp_eventgraph.sentence), 7

sentence (pyknp_eventgraph.event.Event attribute), 8

sentences (pyknp_eventgraph.document.Document
attribute), 7

sentences() (pyknp_eventgraph.eventgraph.EventGraph
property), 7

sentential_complement_event_ids()
(pyknp_eventgraph.argument.Argument prop-
erty), 15

sentential_complement_event_ids()
(pyknp_eventgraph.predicate.Predicate prop-
erty), 13

sid (pyknp_eventgraph.event.Event attribute), 8

sid (pyknp_eventgraph.pas.PAS attribute), 12

sid (pyknp_eventgraph.sentence.Sentence attribute), 7

ssid (pyknp_eventgraph.event.Event attribute), 8

ssid (pyknp_eventgraph.pas.PAS attribute), 12

ssid (pyknp_eventgraph.sentence.Sentence attribute), 7

standard_reps() (pyknp_eventgraph.predicate.Predicate
property), 13

start (pyknp_eventgraph.event.Event attribute), 8

state (pyknp_eventgraph.features.Features attribute),
15

surf (pyknp_eventgraph.relation.Relation attribute), 16

surf() (pyknp_eventgraph.argument.Argument prop-
erty), 15

surf() (pyknp_eventgraph.event.Event property), 11

surf() (pyknp_eventgraph.predicate.Predicate prop-
erty), 13

surf() (pyknp_eventgraph.sentence.Sentence prop-
erty), 8

surf_() (pyknp_eventgraph.event.Event method), 10

surf_with_mark() (pyknp_eventgraph.event.Event
property), 11

surf_with_mark_() (pyknp_eventgraph.event.Event
method), 10

T

tag() (pyknp_eventgraph.argument.Argument prop-
erty), 15

tag() (pyknp_eventgraph.predicate.Predicate prop-
erty), 13

tense (pyknp_eventgraph.features.Features attribute),
15

to_dict() (pyknp_eventgraph.argument.Argument
method), 14

to_dict() (pyknp_eventgraph.component.Component
method), 16

to_dict() (pyknp_eventgraph.document.Document
method), 7

to_dict() (pyknp_eventgraph.event.Event method),
10

to_dict() (pyknp_eventgraph.eventgraph.EventGraph
method), 6

to_dict() (pyknp_eventgraph.features.Features
method), 15

to_dict() (pyknp_eventgraph.pas.PAS method), 12

to_dict() (pyknp_eventgraph.predicate.Predicate
method), 12

to_dict() (pyknp_eventgraph.relation.Relation
method), 16

to_dict() (pyknp_eventgraph.sentence.Sentence
method), 7

to_string() (pyknp_eventgraph.argument.Argument
method), 14

to_string() (pyknp_eventgraph.component.Component
method), 16

to_string() (pyknp_eventgraph.document.Document
method), 7

to_string() (pyknp_eventgraph.event.Event
method), 10

to_string() (pyknp_eventgraph.eventgraph.EventGraph
method), 6

to_string() (pyknp_eventgraph.features.Features
method), 15

`to_string()` (*pyknp_eventgraph.pas.PAS method*), [12](#)
`to_string()` (*pyknp_eventgraph.predicate.Predicate method*), [12](#)
`to_string()` (*pyknp_eventgraph.relation.Relation method*), [16](#)
`to_string()` (*pyknp_eventgraph.sentence.Sentence method*), [7](#)
`type()` (*pyknp_eventgraph.predicate.Predicate property*), [13](#)
`type_` (*pyknp_eventgraph.predicate.Predicate attribute*), [12](#)