pyknp-eventgraph

Kurohashi-Kawahara Lab, Kyoto University

Mar 30, 2021

REFERENCES

CHAPTER

ONE

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.

CHAPTER

TWO

REQUIREMENTS

- Python 3.6 or later
- pyknp
- graphviz

CHAPTER

THREE

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**(*evg*, *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.

CHAPTER

FOUR

AUTHOR/CONTACT

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

- Hirokazu Kiyomaru

4.1 License

CHAPTER

FIVE

INDICES AND TABLES

- genindex
- modindex
- search

PYTHON MODULE INDEX

p

INDEX

I

L

M

N

O

P

R

S

T