
pyknp Documentation

Author

Aug 15, 2023

Contents

1	About	1
2	Requirements	3
3	Install pyknp	5
4	A Simple Explanation of JUMAN++/KNP	7
5	Usage	9
6	Documents	15
6.1	JUMAN module	15
6.2	KNP module	18
6.3	Usage Details and Tips	27
7	Author/Contact	31
8	Indices and Tables	33
	Python Module Index	35
	Index	37

Chapter 2

Requirements

- **Python**
 - Verified Versions: 2.7.15, 3.5.6, 3.6.6.
- **日本語形態素解析器 JUMAN++ (JUMAN)**
 - JUMAN++はJUMANの発展版で、日本語の形態素解析を行う。
 - English pages of JUMAN++ (Japanese Morphological Analyzer): EN_JUMAN++ (EN_JUMAN)
- **日本語依存構文解析器 KNP**
 - English page of KNP (Japanese Dependency and Case Structure Analyzer): EN_KNP

Chapter 3

Install pyknp



A Simple Explanation of JUMAN++/KNP

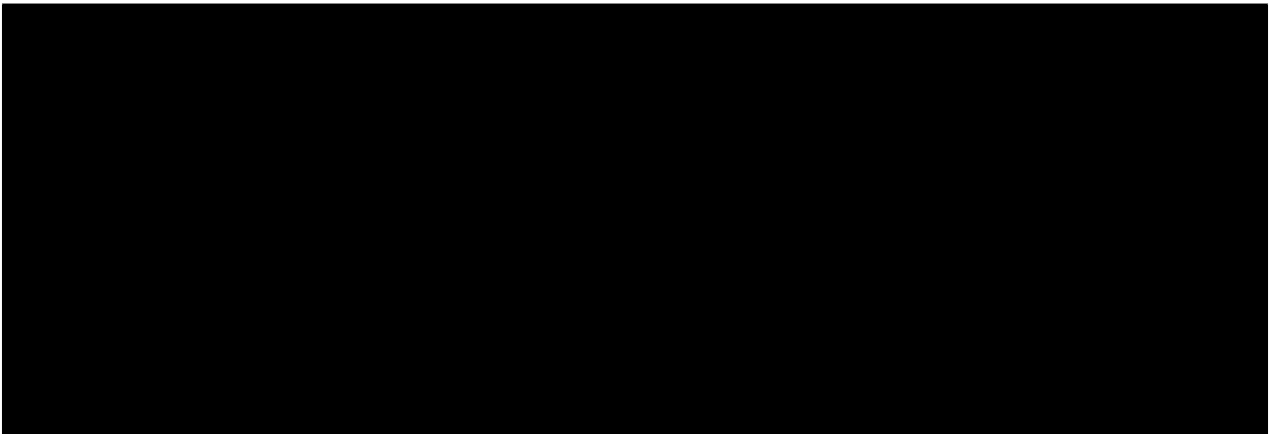
Chapter 5

Usage

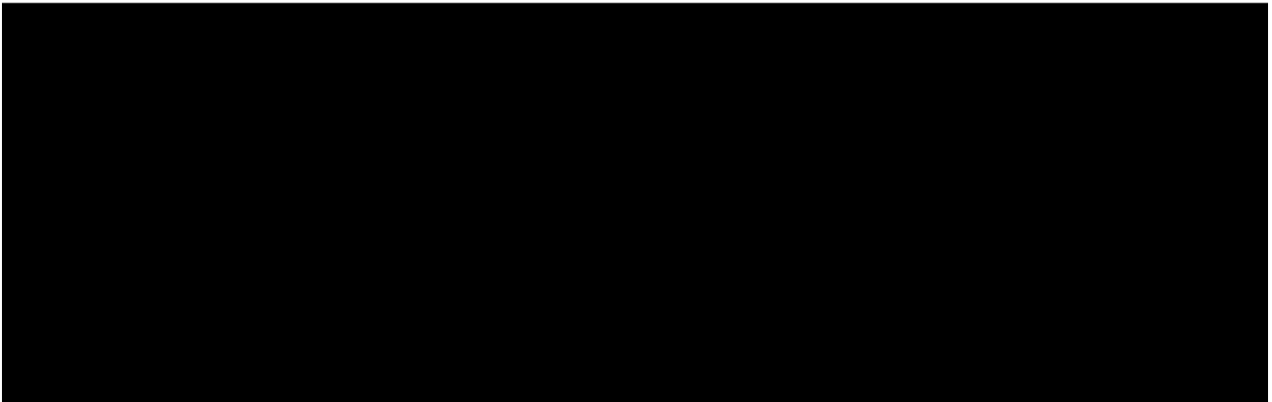
èlşãAŮãAĐä;£ãAĐæŮzãAřãAŞãAqãCLãCŞşãRCçĚğãAŮãAęãARãAăãAŤãAĐãĂĆ *Usage Details and Tips*
éŮcæŤřãCĐãCřãCřãCzãAøèlşçřřãAřäyNèÍŸãCLãCŋãCěãCqãCşãCŁãCCãRCçĚğãAŮãAęãARãAăãAŤãAĐãĂĆ

JUMAN++

JUMAN++ãAôãõşèqŇäĹŇ



pyknpãCŞçŤİãAĐãAşşèğçædŘãCŮãCŋãCřãCřãCă



ãCŮãCŋãCřãCřãCăãAôãGzãLZçŤŘædIJ

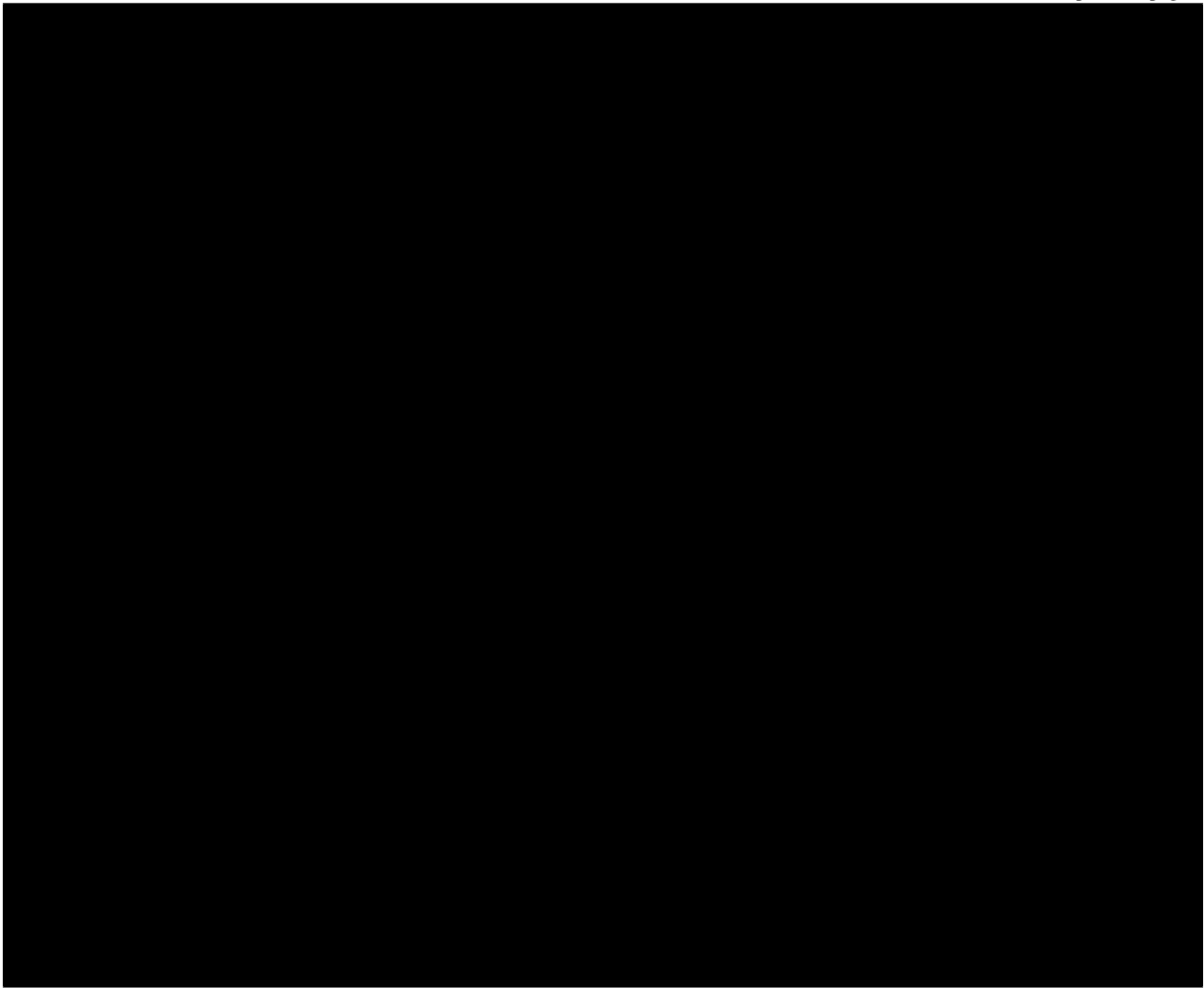
KNP

KNPãAqôãðšèqÑä;ÑãĀĈãGžãŁZãAqôèĭqãAŁæŨžãAŁræñqãAqôURLăĈŠăĤĈĈçĖğăĀĈ
http://cr.fvcrcl.i.nagoya-u.ac.jp/~sasano/KNP/format.html

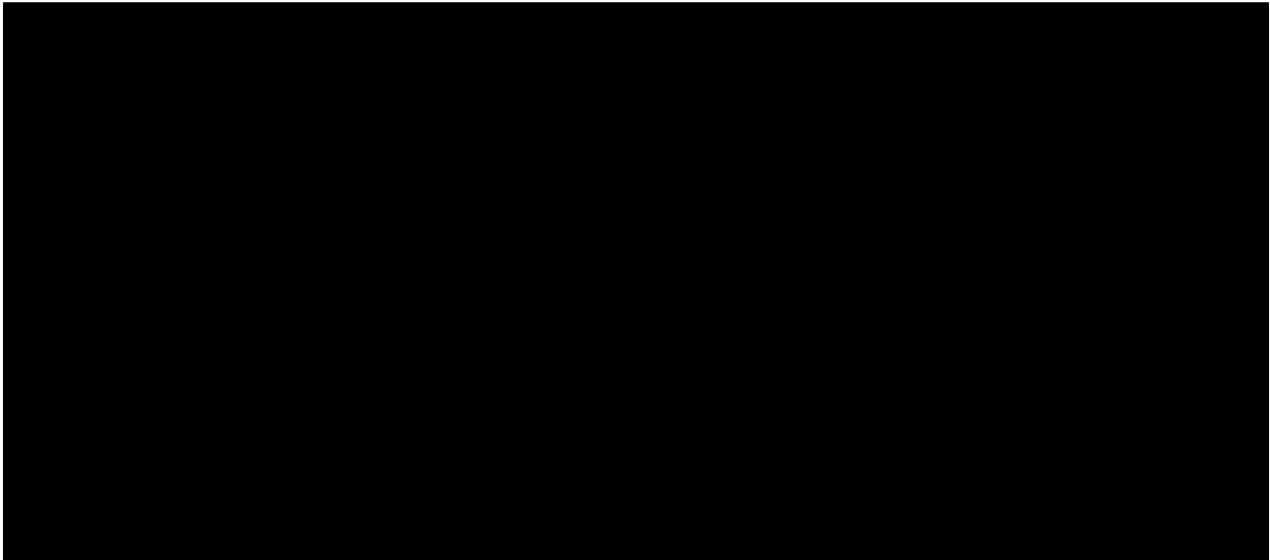
Figure 1

(continues on next page)

(continued from previous page)



pyknpãĆŠçŤĺãAđãAşèğçæđŘãČŮãČŋãČřãČĺãČă



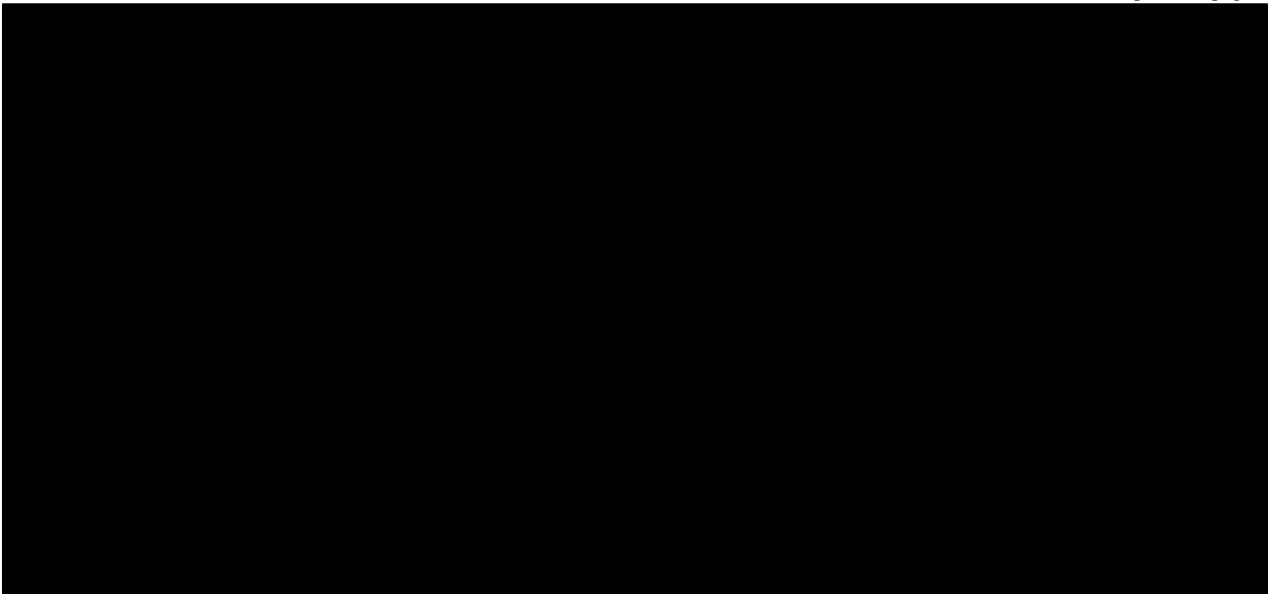
(continues on next page)

(continued from previous page)

ãĈŮãĈŋãĈřãĈřãĈăãAŁăGżăŁŻçtŔæđIJ

WikipediaaDraNY CUSaijjazcealealelY azcealealelY:ayNer1/aAUaCCaANaCC, azcealealelY:ayNer1/aAUaCCaANaCC

(continued from previous page)



Python2çşzãAõääfãŘĹ

pyknpãAğãAřãAžãAęãAõãĚãĜãĹZãAğUnicodeãĈŠãĈşãôZãAŮãAęãAŁãĈĹãĂAãAĹãĈNãžěãďŮãAõæŮĜãŋŮãĈşãĈjãĈĹãAňã
ãAĹãAõãAşãĈAPython2çşzãAğãAřãĂAÿLeĹŸãAõãĈŮãĈŋãĈřãĈĹãĈããĚSéãŋãAõÿNëĹŸãAõ2ëãNãAŋNãŁĚëëAãAğãAĈãĈNãĂĈ



Documents

```
class pyknp.juman.juman.Juman(command='jumanpp', server=None, port=32000, timeout=30, option="",
                                rcfile="", ignorepattern="", pattern='^EOS$', jumanpp=True,
                                multithreading=False)
```

â;çæĚŇctăèğçæđŘăZÍ JUMAN ăĈŠ Python ăȦŇăĈĹăĹĭçȚĭăȦŽăĈŇăȦŞăĈȦăȦőăĈcăĈȳăĈcăĈijăĈñ

- **command** (*str*) – JUMANāĀōāōšēāŃāĈšāĈđāĈšāĈĹ
- **option** (*str*) – JUMANēğĉæđŘāĈĹāĈŮāĈūāĈğāĈš (āĈřāĈĚāĈĈāĈĹāĈāĭĲ -s, āĈšāĈĭjāĈĀāžĚ –beam <int>)
- **rcfile** (*str*) – JUMANēĹāōŽāĈŤāĈqāĈđāĈñāĀŷāĀōāĈšāĈž
- **pattern** (*str*) – JUMANāĠžāĹžāĀōĉĴĈñřēĹŷāŘŮ
- **juanpp** (*bool*) – JUMAN++āĈšĈřĹāĀđāĈŃāĀŃJUMANāĈšĈřĹāĀđāĈŃāĀŃāĈcommandāĈšæŃĠāōžāĀŮā

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

ǣĒēāLZæŨĜāņŪāLŪāAnār; āAŪāAeā; cæEñCtǣēğcædRǣCSèaNǣAĐāĂAāAIāAōJumanāGžāLZçTŘædIJāĆSèfTǣAZ

Returns

```
juman(input_str, juman_format=0)
analysiséŮćæTřãAĺăŘŇăAŸ
```

āĖēāŁZēŪĠāŋŪāŁŪāĤāāŕġāĤŪāAqāġācēĖŅçġāēġçædŘāĤŪāĀĤāAqōçġŘædĪJāČŠ
āČġāČŪāČYāČġāČŕāČĹāĀġāĤŪāAēēĤĤāAZ

Parameters

- **input_str** (*str*) – æŨĜăĆŠëǎİAŻæŨĜǻŦŰăĹŮ
- **juman_format** (*JUMAN_FORMAT*) – JumanăĂōlatticeăĠzăĹZă;ćaijŘ

Returns aĳcæĚNçřřăĽŮăĆłăČŮăĆŸăĆğăĆřăČĹ

Return type *MList*

```
result(input_str, juman_format=0)
```

JumanăGžǎŁŻçŧRædIJãAñârĩ;ãAŮãAęãĂAãAĭãAőçŧRædIJãČŠ MList ãĆlãČŮãCÿãCğãCřãČLãAĭãAŮãAęëƒTãAZ

Parameters

- **input_str** (*str*) – JumanǎĠZ̧ŁŻçŦŘæđIJ
- **juman_format** (*JUMAN_FORMAT*) – JumanǎAłatticeǎĠZ̧ŁŻȳ;ćaijR

Returns `âĶĳĉæĔĤĦġĤăăĹŮăĈĤăĈŮăĈŷăĈğăĈŕăĈĹ`

Return type *MList*

6.1.1 Morpheme modules

MorphemeList module

```
class pyknp.juman.mlist.MList(spec=",",juman_format=0)
```

Bases: object

ãAĈãĈNæŨĜãAñéŨćãAŻãĈNă;ćæĔNċťăăĽŨãĈŚăĔIæÑAãAŻãĈNăĈlăĈŨãĈŸăĈĝăĈrăĈĽIijŎ

```
__init__(spec="", juman_format=0)
```

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

push_mrph(*mrph*)

set_readonly()

spec()

ǎj cǎĖ Nčtǎǎ Lŭǎ Aǒǎ Ėlǎ Ũ Gǎnj Ŭǎ Lŭǎ ČSěǎ TǎA Žijj Ő Juman ā Añǎ ĆLǎ ĆNǎ Ğzǎ LZǎ AÍǎ RÑǎ Aÿǎ ;cǎ jRǎ Aǒçǎ ŘǎedIJǎ AÑǎ Ŭǎ Ć

new_spec()

mrph_list()

äİİæÑĀāZăCŇăĬlăAeăAőă;ćăĚŇćăăAőăČlăČzăČĹăĆšěŤăĂZıııŎ

Returns `â;çæĖÑçtããĈlãĈŨãĈyãĈğãĈrãĈĹMorphemeãAõãĈlãĈzãĈĹ`

Return type list of Morpheme

Morpheme module

```
class pyknp.juman.morpheme.Morpheme(spec, mrph_id=None, juman_format=0)
```

Bases: object

ǎ;ćăĖŇčřãĀOőăRĎđłœČĚăăšăĆŠăŁİăÑĀăZăĆNăĆłăČŮăĆȳăĆğăĆrăČĹiiǾ

Parameters

- **spec** (*str*) – JUMAN/KNPăĜžăĹŽ
- **mrph_id** (*int*) – âĭćæĔŇċťăĪD
- **juman_format** (*JUMAN_FORMAT*) – JumanăA6latticeăĜžăĹŽăĭćăĭĴ

mrph_id
 âĳċæĔŃċřăID
Type int

mrph_index
 mrph_idăAĳăŃăŔăŤăAĲ
Type int

doukei
Type list

midasi
 ěċŃăĠăŤăăAĲ
Type str

yomi
 ěĳăăAĲ
Type str

genkei
 âŖŖăĳċ
Type str

hinsi
 âŖAĲěĳđ
Type str

hinsi_id
 âŖAĲěĳđID
Type int

bunrui
 âŖAĲěĳđċřăĲĲĲăđ
Type str

bunrui_id
 âŖAĲěĳđċřăĲĲĲăđID
Type int

katuyou1
 æĳċŖĲăđŃ
Type str

katuyou1_id
 æĳċŖĲăđŃID
Type int

katuyou2
 æĳċŖĲăĳċ
Type str

katuyou2_id
 æĳċŖĲăĳċID

Type int

imis
æĎŘăŜşæĈĖăăś

Type str

fstring
çřăæĂğæĈĖăăś

Type str

repname
ăžčëąłëąłëŸ

Type str

ranks
ăĈłăĈĖăĈćăĈzăĂğăĂőăĈłăĈşăĈř

Type set[int]

span
ăĳćæĔŇçřăăĂőăĳŊçĳő (éŮŇăğŇăĳŊçĳő, çřĈăžĖăĳŊçĳő), JUMANăĜžăĹŽăĳćăĳŖăăŇăĈłăĈĖăĈćăĈzăĳćăĳŖăăĂőăăřăĹăĂőăă

Type tuple

__init__ (*spec, morph_id=None, juman_format=0*)
Initialize self. See help(type(self)) for accurate signature.

push_doukei (*morph*)

repnames ()
ăĳćæĔŇçřăăĂőăăžčëąłëąłëŸĳĳĹăŽŮăŸğăĂğăĂŇăĂĈăĈŇăăřăĹăĂřăăŇăăŇăăĂğăĂĉçřĳĳĳĹăĈŞëĤăĂžĳĳő

ReturnsăĳćæĔŇçřăăĂőăăžčëąłëąłëŸăŮĜăŋŮăĹŮ

Return type str

spec ()

new_spec (*prev_morph_id=None, span=None*)

6.2 KNP module

```
class pyknp.knp.knp.KNP(command='knp', server=None, port=31000, timeout=60, option='-tab', rcfile='',
    pattern='EOS', jumancommand='jumanpp', jumanrcfile='', jumanoption='',
    jumanpp=True, multithreading=False)
```

Bases: object

KNPăĈŞçŤĹăĂĎăĂęăğŇăŮĜëğçæĎŘăĈŞëąŇăĂĖĹ/KNPăĂőëğçæĎŘçřĹăĈŞëĳăĂĹăŮăĈŇăĈćăĈŸăĈëăĈĳăĈŇă

Parameters

- **command** (*str*) – KNPăĈşăĈĎăĈşăĈĹ
- **option** (*str*) – KNPëğçæĎŘăĈłăĈŮăĈăĈğăĈş (elşçřëğçæĎŘçřĹăĈŞăĜžăĹŽăĂžăĈŇătabăĂřăĤĖăăĹăĂĈ çĴĂçŤëăĈzçĖğăĴĴëğçæĎŘăĈŞëąŇăĂĖ -anaphora, æăĳĳëğçæĎŘăĈŞëąŇăĈŖăĂžăğŇăŮĜëğçæĎŘăĂőăĂĹăĈŞëąŇăĂĖ -dpndăĂłăĂĴ)
- **rcfile** (*str*) – KNPëĴŋăőžăĈŤăĈăăĈĎăăĴăăĂőăĈŞăĈž
- **pattern** (*str*) – KNPăĜžăĹŽăĂőçřĈçŋŖëĴŸăŖă

- **jumancommand** (*str*) – JUMANāĆšāČďāČšāČĽ
- **jumanrcfile** (*str*) – JUMANēĪŋāōZāČŤāČāāČďāČŋāĀŷāĀōāČŚāČž
- **jumanpp** (*bool*) – JUMAN++āČŚŸŤĪāĀďāČŇāĀŇJUMANāČŚŸŤĪāĀďāČŇāĀŇ
- **multithreading** (*bool*) – ěğçæďŘāČŚāČāāČďāČšāČžāČŋāČČāČĽāžēādŮāĀŇāČĽēāŇāĀĒāŘrèČ;æĀğāĀŇāĀČ

__init__ (*command='knp', server=None, port=31000, timeout=60, option='-tab', rcfile='', pattern='EOS', jumancommand='jumanpp', jumanrcfile='', jumanoption='', jumanpp=True, multithreading=False*)

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

knp (*sentence*)

parseēŮćæŤřāĀĪāŖŇāĀŸ

parse (*sentence, juman_format=0*)

āĚēāĽZāĀŤāČŇāĀšæŮĜāŋŮāĽŮāĀŋāŕĴāĀŮāĀęā;ćæĒŇŸtāēğçæďŘāĀĪāęŇæŮĜēğçæďŘāČŚēāŇāĀďāĀĀæŮĜŸŕĀāĽŮāČĪāČŮāČŷāČğāČŕāČĽāČŚēĽŤā

Parameters

- **sentence** (*str*) – æŮĜāČŚēāĪāĀŽæŮĜāŋŮāĽŮ
- **juman_format** (*JUMAN_FORMAT*) – JumanāĀōlatticeāĜžāĽZā;ćāĪŖ

Returns æŮĜŸŕĀāĽŮāČĪāČŮāČŷāČğāČŕāČĽ

Return type *BList*

parse_juman_result (*juman_str, juman_format=0*)

JUMANāĜžāĽZŸŤŘædĪJāĀŋāŕĴāĀŮāĀęæęŇæŮĜēğçæďŘāČŚēāŇāĀďāĀĀæŮĜŸŕĀāĽŮāČĪāČŮāČŷāČğāČŕāČĽāČŚēĽŤā

Parameters

- **juman_str** (*str*) – āĀČāČŇæŮĜāĀŋēŮćāĀŽāČŇJUMANāĀōāĜžāĽZŸŤŘædĪJ
- **juman_format** (*JUMAN_FORMAT*) – JumanāĀōlatticeāĜžāĽZā;ćāĪŖ

Returns æŮĜŸŕĀāĽŮāČĪāČŮāČŷāČğāČŕāČĽ

Return type *BList*

reparse_knp_result (*knp_str, juman_format=0*)

KNPāĜžāĽZŸŤŘædĪJāĀŋāŕĴāĀŮāĀęāČČāĀĒāŷāĀžęæęŇæŮĜēğçæďŘāČŚēāŇāĀďāĀĀæŮĜŸŕĀāĽŮāČĪāČŮāČŷāČğāČŕāČĽāČŚēĽŤā
KNPāĀōfeatureāČŚāĒŷāŷŮāĀŽāČŇāāŕĽāĀĪāĀĪāĀŋŸŤĪāĀďāČŇāĀČāŷŋēžŋāĀŕparse_juman_resultēŮćæŤřāĀĪāŖŇāĀŸ

Parameters

- **knp_str** (*str*) – āĀČāČŇæŮĜāĀŋēŮćāĀŽāČŇKNPāĀōāĜžāĽZŸŤŘædĪJ
- **juman_format** (*JUMAN_FORMAT*) – JumanāĀōlatticeāĜžāĽZā;ćāĪŖ

Returns æŮĜŸŕĀāĽŮāČĪāČŮāČŷāČğāČŕāČĽ

Return type *BList*

result (*input_str, juman_format=0*)

āĀČāČŇæŮĜāĀŋēŮćāĀŽāČŇKNPēğçæďŘŸŤŘædĪJāČŚæŮĜŸŕĀāĽŮāČĪāČŮāČŷāČğāČŕāČĽāĀŋāđĽæŖZāĀŽāČŇ

Parameters

- **input_str** (*str*) – āĀČāČŇæŮĜāĀŋēŮćāĀŽāČŇKNPāĀōāĜžāĽZŸŤŘædĪJ
- **juman_format** (*JUMAN_FORMAT*) – JumanāĀōlatticeāĜžāĽZā;ćāĪŖ

Returns æŮĜŸŕĀāĽŮāČĪāČŮāČŷāČğāČŕāČĽ

Return type *BList*

6.2.1 Bunsetsu modules

BunsetsuList module

```
class pyknp.knp.blist.BList(spec=",", pattern='EOS', juman_format=0)
```

Bases: object

ãAçCãCÑæÛGãAñéÛcãAžZãCÑæÛGçřAãLŮãCŠšãIæÑAqãAžZãCÑãCłãCŮãCýãCğãCřãCŁ

Parameters

- **spec** (*str*) – KNPǎĜžǎŁŻ
- **pattern** (*str*) – KNPǎĜžǎŁŻǎAőçŦÇćńřèİYǎRũ
- **juman_format** (*JUMAN_FORMAT*) – JumanǎAólatticeǎĜžǎŁŻǎ;ćaijŘ

comment

KNPǎĜžǎŁZǎAńǎAŁǎAŠǎĆNǎǞAǎ#ǎAǎNǎCLǎğNǎAǎǎĆNǎèǎNǎAǎǎZǎǎAǎNǎĆNǎAǎǎǎUǎǎǎUǎǎLǎUǎ

Type str

sid

æŮĜID (KNPǎĜżǎŁŻäyŋãAŹS-ID)

Type str

```
__init__(spec="", pattern='EOS', juman_format=0)
```

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

parse(*spec*)

KNPǎAŁǎGżǎLZǎĆŠèŋǎAŁǎRŪǎĆN

Parameters `spec (str)` – KNPåĞżåŁŻ

```
get_tag_span(tag_id)
```

ǎșžæIJňăŘěãAőă;Nç;óæČĚăăšăĆŠèŁTăȚZ

Parameters `tag_id (int)` – ID

Returns (åşžæIJňăŘěãĄóéŨŇăğŇăĵNçĵó, çťĆăžĚăĵNçĵó)

Return type tuple

push_bnst(*bnst*)

tag_list()

æŮĜăĈŠæĝNăĹŔăĀZăĈNăĖĹăŝzæIJnăŔăăĈăăĈŮăĈyăĈĝăĈŕăĈĹăĈŠăĤăĀZ

Returns åşžæIJñåRẽãĆłăČŮãČÿãĆġãĆřãČĹTagãAõãČłăĆžãČĹ

Return type list of Tag

mrph_list()

æUǦăĈŠæǧNăĹŔăĀZăĈNăĬlă:caeĤNctăăĈlăĈŪăĈyăĈġăĈŕăĈĹăĈŞeŧŤăĀZ

Returns `â;çæĚŇçĩăăĈĩăĈŮăĈyăĈğăĈrăĈĹMorphemeăĂăăĈĩăĈzăĈĹ`

Return type list of Morpheme

bnst_list()

æŮĜăĈŠæġNăĹŔăĂZăĈNăĖĬăŮĜċŕĂăĈĭăĈŮăĈvăĈġăĈŕăĈĬăĈŠăĤăĂZ

Returns æŨĞçřĂăČłăČŨăČÿăČğăČřăČĹBunsetsuăĂăăČłăČźăČĹ

Return type list of Bunsetsu


```

children
    æŋŕãAõæŮĜçfĂăCłăCŮăCÿăCġăCřăCĹăAõăCłăCzăCĹ

    Type list

repname
    æŋçèçĖŘăŇŮăzçèçłèçłèĹŸ (normalized_repnameăAŋăŕŇăAŸ)

    Type str

normalized_repname
    æŋçèçĖŘăŇŮăzçèçłèçłèĹŸ

    Type str

head_repname
    äÿzèçđăzçèçłèçłèĹŸ

    Type str

head_prime_repname
    äÿzèçđăĂZăzçèçłèçłèĹŸ

    Type str

fstring
    featureæCĚăăś

    Type str

__init__(spec, bnst_id=0, juman_format=0)
    Initialize self. See help(type(self)) for accurate signature.

push_morph(mrph)
    æŮŕăAŮăAĎăđ;çæĚŇçřăăCłăCŮăCÿăCġăCřăCĹăCŚăCzăCĈăCĹăAŻăCŇ

push_tag(tag)
    æŮŕăAŮăAĎăđ$zæIJăŋăŘăăCłăCŮăCÿăCġăCřăCĹăCŚăCzăCĈăCĹăAŻăCŇ

set_midasi()
    midasiăCŚăCzăCĈăCĹăAŻăCŇ

spec()
    æŮĜçfĂăAŋăŕçłăŋIJăĂZăCŇKNPăĜzăLZ

mrph_list()
    æŮĜçfĂăCŚăçĖŇăĹŖăAŻăCŇăĚłăçæĚŇçřăăCłăCŮăCÿăCġăCřăCĹăCŚçłŤăAŻ

    Returns æŋçæĚŇçřăăCłăCŮăCÿăCġăCřăCĹMorphemeăAõăCłăCzăCĹ

    Return type list

tag_list()
    æŮĜçfĂăCŚăçĖŇăĹŖăAŻăCŇăĚłăçzæIJăŋăŘăăCłăCŮăCÿăCġăCřăCĹăCŚçłŤăAŻ

    Returns æŋçæIJăŋăŘăăCłăCŮăCÿăCġăCřăCĹTagăAõăCłăCzăCĹ

    Return type list

pstring(string=None)
    draw_treeăAŮăAŞăAĹăANăANăRşăAŤăANăĜzăLZăAŻăCŇæŮĜăŋŮăĹŮăCŚçłŤăAŻ

```

6.2.2 Tag modules

TagList module

class pyknp.knp.tlist.**TList**

Bases: object

ãAÇãCÑæŨGãAñéŨcãAŽãCÑãšžæIJñãRëãLŨãCŠãŁiæÑãAŽãCÑãCłãČŨãCÿãCğãCřãČĹ

__init__()

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

push_tag(tag)

push_morph(morph)

push_synnodes(synnodes)

push_synnode(synnode)

spec()

set_readonly()

draw_tree(fh=None, show_pos=True)

draw_tag_tree(fh=None, show_pos=True)

ãCŁãCřãLŨãAğãŁĩãŃŸéŨcãŁCãCŠæIJĩãğNéĂããAĩãAŨãAęãŁĩçRĩãAŨãAęãĞžãLŽãAŽãCÑĩĩjŮ

sprint_tree()

tag_list()

ãšžæIJñãRëãLŨãCŠæğNæĹRãAŽãCÑãĹĩãšžæIJñãRëãCłãČŨãCÿãCğãCřãČĹãCŠèŁTãAŽ

Returns ãšžæIJñãRëãCłãČŨãCÿãCğãCřãČĹTagãAğãCłãCžãČĹ

Return type list of Tag

Tag module

class pyknp.knp.tag.**Tag**(spec, tag_id=0, juman_format=0)

Bases: object

ãAÇãCÑæŨGãAñéŨcãAŽãCÑãšžæIJñãRëãLŨãCŠãŁiæÑãAŽãCÑãCłãČŨãCÿãCğãCřãČĹ

Parameters

- **spec** (str) – KNPãĞžãLŽ
- **tag_id** (int) – ãšžæIJñãRëID
- **juman_format** (JUMAN_FORMAT) – JumanãAğłatticeãĞžãLŽãĩçãĩjŘ

tag_id

ãšžæIJñãRëID

Type int

midasi

èçÑãĞžãAŨ

Type str

parent

èĕłãAŏăšžæIJňăŘěăČłăČŮăČyăČĝăČřăČĹ

Type *Tag***parent_id**

èĕłãAŏăšžæIJňăŘěID

Type *int***children**

ăŋŘăAŏăšžæIJňăŘěăČłăČŮăČyăČĝăČřăČĹăAŏăČłăČžăČĹ

Type *list***dpndtype**

ăĕČăČĹăĹŮăAŏăŠăČĕăăČďăČŮ

Type *str***fstring**

featureăĈĚăăś

Type *str***repname**

æŋĉĕĕŘăŇŮăžĉĕăĹăĹăĹŸ (normalized_repnameăAňăŘňăAŸ)

Type *str***normalized_repname**

æŋĉĕĕŘăŇŮăžĉĕăĹăĹăĹŸ

Type *str***head_repname**

ăÿzèĭďăžĉĕăĹăĹăĹŸ

Type *str***head_prime_repname**

ăÿzèĭďăĂŽăžĉĕăĹăĹăĹŸ

Type *str***pred_repname**

ċŦĹăĹĂăžĉĕăĹăĹăĹŸ

Type *str***disambiguated_pred_repname**

æĹŽăžŮċŦĹăĹĂăžĉĕăĹăĹăĹŸ

Type *str***features**

ăšžæIJňăŘěăAŏăfeatureăČŠăĹăAŽăFeatureăČłăČŮăČyăČĝăČřăČĹ

Type *Features***pas**ăšžæIJňăŘěăAŇăĕřĕĹďăAŏăăĤăĹăAřăăĚăAŏăăĈĚăăś(PasăČłăČŮăČyăČĝăČřăČĹ),
ăAĹăAĖăAĝăAĹăAĎăăĤăĹăNone**Type** *Pas*

```

__init__(spec, tag_id=0, juman_format=0)
    Initialize self. See help(type(self)) for accurate signature.

push_morph(morph)
    æŮřãAŮãAĐã;ćæĚŇčřããCřãCŮãCŷãCğãCřãCřãCšãCzãCčãCřãAžãCŇ

set_midasi()
    midasiãCšãCzãCčãCřãAžãCŇ

spec()
    âšžæIJňãŘěãAňãř;âĚIJãAžãCŇKNPãĜžãLž

morph_list()
    âšžæIJňãŘěãCšãĜŇãĹãAžãCŇãĚĹã;ćæĚŇčřããCřãCŮãCŷãCğãCřãCřãCšẽřTãAž

    Returns  â;ćæĚŇčřããCřãCŮãCŷãCğãCřãCřMorphemeãAőãCřãCzãCř

    Return type  list

pstring(string=None)
    draw_treeãAŮãAšããAĹãAŇãAňãŘșãAřãAňãĜžãLžãAžãCŇãĚĜãŇŮãĹŮãCšẽřTãAž

get_surface()
    âšžæIJňãŘěãAőẽĚãĜžãAŮãCšẽřTãAž

    Returns  âšžæIJňãŘěãAőẽĚãĜžãAŮ

    Return type  str

```

PAS module

```
class pyknp.knp.pas.Pas(tid=None, result=None)
```

Bases: object

```
ẽřřẽĹdẽãĚæĝŇẽĂããCšæĹsãAĚãCřãCřãCž
```

```
Usage: result = knp.result(knp_result) pas = result.tag_list()[tid].pas #
tidçŤřçŽõãAőãšžæIJňãŘě(ẽřřẽĹd)ãAőẽãĚæĝŇẽĂã
```

cfid

```
æãijãCřãCňãCijãCãID (ãĹŇ: âĚIJẽãŇãAĚ/ãAĹãAšããAřãAĚ:ãŇŤ10ãĂĪ)
```

Type str

arguments

```
æãijãAĹẽãĚãCšãř;âĚIJãžŸãAšãAšẽĹđæŽŷ {case: [Argument, ..]} keyãAřæãijãCšẽãĹãAžæŮĜãŇŮãĹŮ,
valueãAřArgumentãCřãCŮãCŷãCğãCřãCřãAőãCřãCzãCčãCřãCšẽřTãAž
```

Type dict of (case, list of Argument)

```
__init__(tid=None, result=None)
```

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

is_valid()

get_arguments(case)

```
æŇĜãõŽãAŮãAšæãijãAőãŘĐẽãĚãAřãAĹãAňãžçẽãĹẽãĹẽŸãAőẽĚŇãĹŮãCšẽřTãAž
```

```
Usage: result = knp.parse(âĚIJçãŤçĹũẽĂĚãAňãAřãCŇãĂĪ) print(result.tag_list()[2].midasi)
>>  ãAřãCŇ print(result.tag_list()[2].pas.get_arguments(âĚIJãCňãĂĪ)) >>  [ArgRep-
name(repname=âĂžẽĂĚ/ãAŮãCčãĂž, tid_list=1)]
```

Parameters `case (str)` – æäijãAõæŮĜãŋŮãĹŮ

Returns éãĚãAõäzçèãĹèãĹèĹŸãĈŠæäijçŧNãAŮãA\$namedtupleãAğãAĈãĈN ArgRepname
ãAõãĈĹãĈãĈĹ

Return type list

get_orig_result()

class pyknp.knp.pas.**Argument**(*sid=None, tid=None, eid=None, midasi="", flag=None, sdist=None*)

Bases: object

éãĚãAñèŮãAŻãĈNãĈĚããããĈŠãĹãNããAŻãĈNãĈĹãĈŮãĈŸãĈğãĈřãĈĹ

èĹãAŮãAĹãAřãNèĹŸãĈŽãĈijãĈŸãAõããNãäijèçAçŧããAŧããNãAõèĹŸèřæŮzæŧãĈŠãĹĈçĚğ http:
//nlp.ist.i.kyoto-u.ac.jp/index.php?KNP%2F%E6%A0%BC%E8%A7%A3%E6%9E%90%E7%B5%90%
E6%9E%9C%E6%9B%B8%E5%BC%8F

sid

æŮĜID

Type str

tid

ãŧžæIJñãĹĚID

Type int

eid

Entity ID

Type int

midasi

èãĹèĹŸ

Type str

flag

ãĈŧãĈĹãĈř (C, N, O, D, E, U)

Type str

sdist

èřřèĹdãAõãjŧæŮĜãĹNãAŹ

Type int

__init__(*sid=None, tid=None, eid=None, midasi="", flag=None, sdist=None*)

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

Feature module

class pyknp.knp.features.**Features**(*spec, splitter='><', ignore_first_character=True*)

Bases: dict

featureæĈĚããããĈŠãĹãNããAŻãĈNãĈĹãĈŮãĈŸãĈğãĈřãĈĹ

featureæĈĚããããAñãĹãĹãAĹ;ãĈNãĈNãĈĹãĈřãĈŠãĈŠãĈijãĈŽãAŮããAëçđæŽŸãçãijãĹãAñãAŻãĈNãĈĹ ex.
ãĹIJ<æŋçèçRãNŮãzçèãĹèãĹèĹŸ:éAĚãĈNãĈN/ãAĹãAĹãĈNãĈN>ãĹI -> {ãĹIJæŋçèçRãNŮãzçèãĹèãĹèĹŸãĹI:
ãĹIJéAĚãĈNãĈN/ãAĹãAĹãĈNãĈNãĹI}

```
__init__(spec, splitter='><', ignore_first_character=True)
    Initialize self. See help(type(self)) for accurate signature.

property pas
```

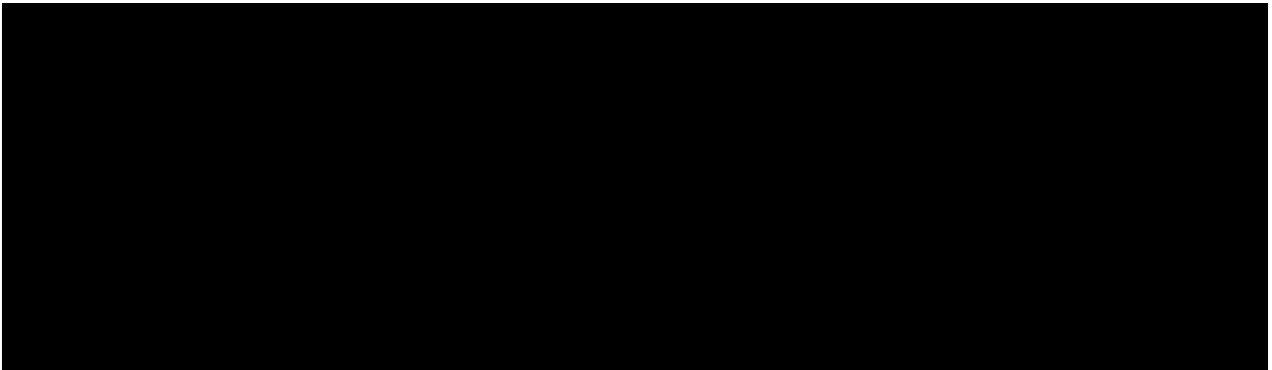
6.2.3 Drawing Tree module

```
pyknp.knp.drawtree.draw_tree(leaves, fh=None, show_pos=True)
pyknp.knp.drawtree.sprint_tree(leaves, show_pos=True)
```

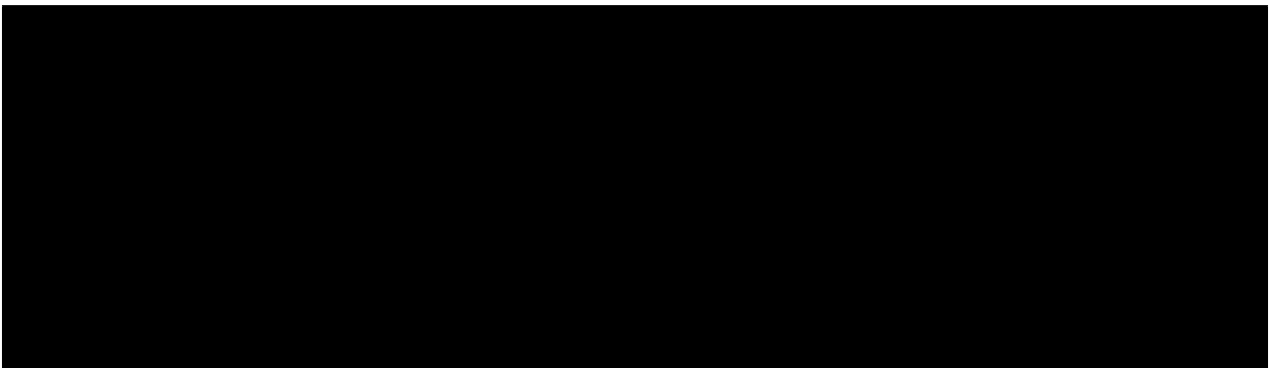
6.3 Usage Details and Tips

6.3.1 äžĹāĆAæžŮāĆŽāAŮāA§JUMAN/KNPèğçædŘçťŘædIJăČťăCqăĆdăČňăĆŠăĹĭçťĹăAŽăĆNæ

JUMAN++āĈĎKNPăAřæšžăAŮăAçéĂşăAřăAĹăAĎăAşăĈăăĂAăđğéĜRăAřăĈEăĈŋăĈzăĈĹăĈŞèğçædŘăAŽăĆNéŽžăAř
 juman.analysiséŮćæťřăĈĎknp.parseéŮćæťřăĈŞçťĹăAĎăAçăăôşăŋăăAăşăăAşăăAňăăćæĚŇçťăèğçædŘăĈĎăğŇæŮĜèğçædŘăĈŞèăŇă
 äžĹāĆAJUMAN++/KNPăĈŞéAĭçťĹăAŮăAşèğçædŘæýĹăAřăĈťăCqăĆdăČňăĆŞçťĹăĎŘăAŮăAşăĈŇăĈŞèĭŋăAřèçĭjăĈĂæŮžăAňèĹ
 Juman/KNPăĈĹăĈŮăĈŷăĈğăĈřăĈĹăAřresultéŮćæťřăAňăĹăŮĜăAŽăAĎăAçèğçædŘçťŘædIJăĈŞæŷăăAŽăAşăAĹăAğăĂAçĭŋăAřèçĭjă
 JUMAN++



KNP

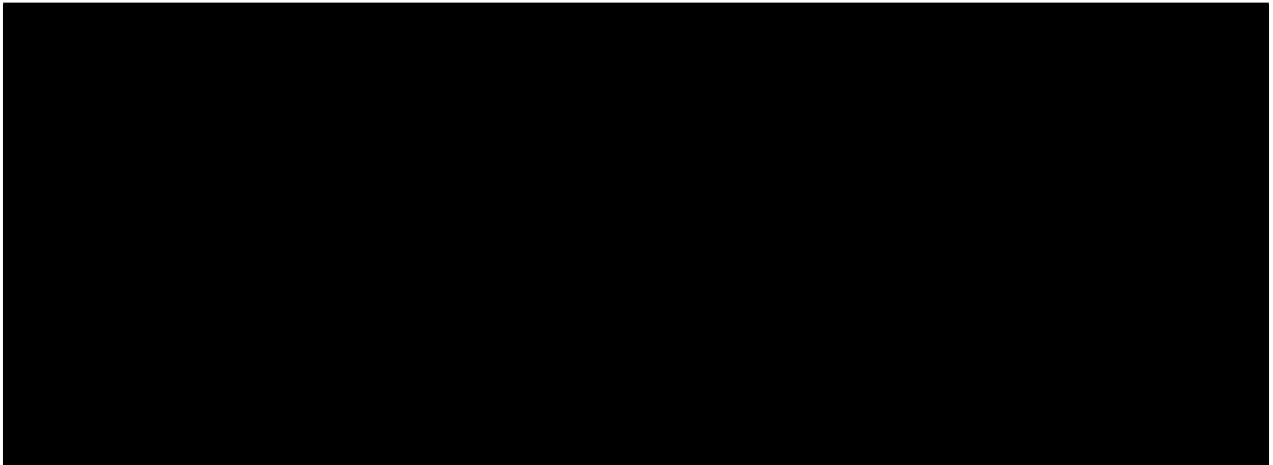


äƒCăĆŁăRŮăAŞéŮćăƒCăAŕæIJlæğNéĂăăĆšăAłăAŻăĂĆ

- [illegible]

èřrèldeáĚæğNéĂăăAíãAřãĂÄä;TãAÑãAíãAEãAŮãAŞãĂÄãAíãAĎãAEèřrèlđãAíãAíãAőéãĚãAÑãĈlãAíãĈNægNé/

Chapter 6. Documents



Chapter 7

Author/Contact

äžñéČ;ädğâηę éžŠæIÑăČzæšşăŎşçăŤçlŭåód (contact@nlp.ist.i.kyoto-u.ac.jp)

- John Richardson, Tomohide Shibata, Yuta Hayashibe, Tomohiro Sakaguchi

Chapter 8

Indices and Tables

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

Python Module Index

p

- `pyknp.juman.mlist`, 16
- `pyknp.juman.morpheme`, 16
- `pyknp.knp.blist`, 20
- `pyknp.knp.bunsetsu`, 21
- `pyknp.knp.drawtree`, 27
- `pyknp.knp.features`, 26
- `pyknp.knp.pas`, 26
- `pyknp.knp.tag`, 23
- `pyknp.knp.tlist`, 23

Index

Symbols

`__init__()` (*pyknp.juman.juman.Juman* method), 15
`__init__()` (*pyknp.juman.mlist.MList* method), 16
`__init__()` (*pyknp.juman.morpheme.Morpheme* method), 18
`__init__()` (*pyknp.knp.blist.BList* method), 20
`__init__()` (*pyknp.knp.bunsetsu.Bunsetsu* method), 22
`__init__()` (*pyknp.knp.features.Features* method), 26
`__init__()` (*pyknp.knp.knp.KNP* method), 19
`__init__()` (*pyknp.knp.pas.Argument* method), 26
`__init__()` (*pyknp.knp.pas.Pas* method), 25
`__init__()` (*pyknp.knp.tag.Tag* method), 24
`__init__()` (*pyknp.knp.tlist.TList* method), 23

A

`all()` (*pyknp.knp.blist.BList* method), 21
`analysis()` (*pyknp.juman.juman.Juman* method), 15
`Argument` (class in *pyknp.knp.pas*), 26
`arguments` (*pyknp.knp.pas.Pas* attribute), 25

B

`BList` (class in *pyknp.knp.blist*), 20
`bnst_id` (*pyknp.knp.bunsetsu.Bunsetsu* attribute), 21
`bnst_list()` (*pyknp.knp.blist.BList* method), 20
`bunrui` (*pyknp.juman.morpheme.Morpheme* attribute), 17
`bunrui_id` (*pyknp.juman.morpheme.Morpheme* attribute), 17
`Bunsetsu` (class in *pyknp.knp.bunsetsu*), 21

C

`cfid` (*pyknp.knp.pas.Pas* attribute), 25
`children` (*pyknp.knp.bunsetsu.Bunsetsu* attribute), 21
`children` (*pyknp.knp.tag.Tag* attribute), 24
`comment` (*pyknp.knp.blist.BList* attribute), 20

D

`disambiguated_pred_repname` (*pyknp.knp.tag.Tag* attribute), 24
`doukei` (*pyknp.juman.morpheme.Morpheme* attribute), 17
`dpndtype` (*pyknp.knp.tag.Tag* attribute), 24
`draw_bnst_tree()` (*pyknp.knp.blist.BList* method), 21
`draw_tag_tree()` (*pyknp.knp.blist.BList* method), 21

`draw_tag_tree()` (*pyknp.knp.tlist.TList* method), 23
`draw_tree()` (in module *pyknp.knp.drawtree*), 27
`draw_tree()` (*pyknp.knp.blist.BList* method), 21
`draw_tree()` (*pyknp.knp.tlist.TList* method), 23

E

`eid` (*pyknp.knp.pas.Argument* attribute), 26

F

`Features` (class in *pyknp.knp.features*), 26
`features` (*pyknp.knp.tag.Tag* attribute), 24
`flag` (*pyknp.knp.pas.Argument* attribute), 26
`fstring` (*pyknp.juman.morpheme.Morpheme* attribute), 18
`fstring` (*pyknp.knp.bunsetsu.Bunsetsu* attribute), 22
`fstring` (*pyknp.knp.tag.Tag* attribute), 24

G

`genkei` (*pyknp.juman.morpheme.Morpheme* attribute), 17
`get_arguments()` (*pyknp.knp.pas.Pas* method), 25
`get_clause_starts()` (*pyknp.knp.blist.BList* method), 21
`get_orig_result()` (*pyknp.knp.pas.Pas* method), 26
`get_surface()` (*pyknp.knp.tag.Tag* method), 25
`get_tag_span()` (*pyknp.knp.blist.BList* method), 20

H

`head_prime_repname` (*pyknp.knp.bunsetsu.Bunsetsu* attribute), 22
`head_prime_repname` (*pyknp.knp.tag.Tag* attribute), 24
`head_repname` (*pyknp.knp.bunsetsu.Bunsetsu* attribute), 22
`head_repname` (*pyknp.knp.tag.Tag* attribute), 24
`hinsi` (*pyknp.juman.morpheme.Morpheme* attribute), 17
`hinsi_id` (*pyknp.juman.morpheme.Morpheme* attribute), 17

I

`imis` (*pyknp.juman.morpheme.Morpheme* attribute), 18
`is_valid()` (*pyknp.knp.pas.Pas* method), 25

J

`Juman` (class in *pyknp.juman.juman*), 15

juman() (*pyknp.juman.juman.Juman method*), 15
juman_lines() (*pyknp.juman.juman.Juman method*), 15

K

katyou1 (*pyknp.juman.morpheme.Morpheme attribute*), 17
katyou1_id (*pyknp.juman.morpheme.Morpheme attribute*), 17
katyou2 (*pyknp.juman.morpheme.Morpheme attribute*), 17
katyou2_id (*pyknp.juman.morpheme.Morpheme attribute*), 17
KNP (*class in pyknp.knp.knp*), 18
knp() (*pyknp.knp.knp.KNP method*), 19

M

midasi (*pyknp.juman.morpheme.Morpheme attribute*), 17
midasi (*pyknp.knp.bunsetsu.Bunsetsu attribute*), 21
midasi (*pyknp.knp.pas.Argument attribute*), 26
midasi (*pyknp.knp.tag.Tag attribute*), 23
MList (*class in pyknp.juman.mlist*), 16
module
 pyknp.juman.mlist, 16
 pyknp.juman.morpheme, 16
 pyknp.knp.blist, 20
 pyknp.knp.bunsetsu, 21
 pyknp.knp.drawtree, 27
 pyknp.knp.features, 26
 pyknp.knp.pas, 25, 26
 pyknp.knp.tag, 23
 pyknp.knp.tlist, 23
Morpheme (*class in pyknp.juman.morpheme*), 16
mrph_id (*pyknp.juman.morpheme.Morpheme attribute*), 16
mrph_index (*pyknp.juman.morpheme.Morpheme attribute*), 17
mrph_list() (*pyknp.juman.mlist.MList method*), 16
mrph_list() (*pyknp.knp.blist.BList method*), 20
mrph_list() (*pyknp.knp.bunsetsu.Bunsetsu method*), 22
mrph_list() (*pyknp.knp.tag.Tag method*), 25

N

new_spec() (*pyknp.juman.mlist.MList method*), 16
new_spec() (*pyknp.juman.morpheme.Morpheme method*), 18
normalized_repname (*pyknp.knp.bunsetsu.Bunsetsu attribute*), 22
normalized_repname (*pyknp.knp.tag.Tag attribute*), 24

P

parent (*pyknp.knp.bunsetsu.Bunsetsu attribute*), 21

parent (*pyknp.knp.tag.Tag attribute*), 23
parent_id (*pyknp.knp.bunsetsu.Bunsetsu attribute*), 21
parent_id (*pyknp.knp.tag.Tag attribute*), 24
parse() (*pyknp.knp.blist.BList method*), 20
parse() (*pyknp.knp.knp.KNP method*), 19
parse_juman_result() (*pyknp.knp.knp.KNP method*), 19
Pas (*class in pyknp.knp.pas*), 25
pas (*pyknp.knp.features.Features property*), 27
pas (*pyknp.knp.tag.Tag attribute*), 24
pred_repname (*pyknp.knp.tag.Tag attribute*), 24
pstring() (*pyknp.knp.bunsetsu.Bunsetsu method*), 22
pstring() (*pyknp.knp.tag.Tag method*), 25
push_bnst() (*pyknp.knp.blist.BList method*), 20
push_doukei() (*pyknp.juman.morpheme.Morpheme method*), 18
push_mrph() (*pyknp.juman.mlist.MList method*), 16
push_mrph() (*pyknp.knp.bunsetsu.Bunsetsu method*), 22
push_mrph() (*pyknp.knp.tag.Tag method*), 25
push_mrph() (*pyknp.knp.tlist.TList method*), 23
push_synnode() (*pyknp.knp.tlist.TList method*), 23
push_synnodes() (*pyknp.knp.tlist.TList method*), 23
push_tag() (*pyknp.knp.bunsetsu.Bunsetsu method*), 22
push_tag() (*pyknp.knp.tlist.TList method*), 23
pyknp.juman.mlist
 module, 16
pyknp.juman.morpheme
 module, 16
pyknp.knp.blist
 module, 20
pyknp.knp.bunsetsu
 module, 21
pyknp.knp.drawtree
 module, 27
pyknp.knp.features
 module, 26
pyknp.knp.pas
 module, 25, 26
pyknp.knp.tag
 module, 23
pyknp.knp.tlist
 module, 23

R

ranks (*pyknp.juman.morpheme.Morpheme attribute*), 18
reparse_knp_result() (*pyknp.knp.knp.KNP method*), 19
repname (*pyknp.juman.morpheme.Morpheme attribute*), 18
repname (*pyknp.knp.bunsetsu.Bunsetsu attribute*), 22
repname (*pyknp.knp.tag.Tag attribute*), 24
repnames() (*pyknp.juman.morpheme.Morpheme method*), 18
result() (*pyknp.juman.juman.Juman method*), 16

`result()` (*pyknp.knp.knp.KNP method*), 19

S

`sdist` (*pyknp.knp.pas.Argument attribute*), 26

`set_midasi()` (*pyknp.knp.bunsetsu.Bunsetsu method*), 22

`set_midasi()` (*pyknp.knp.tag.Tag method*), 25

`set_readonly()` (*pyknp.juman.mlist.MList method*), 16

`set_readonly()` (*pyknp.knp.blist.BList method*), 20

`set_readonly()` (*pyknp.knp.tlist.TList method*), 23

`sid` (*pyknp.knp.blist.BList attribute*), 20

`sid` (*pyknp.knp.pas.Argument attribute*), 26

`span` (*pyknp.juman.morpheme.Morpheme attribute*), 18

`spec()` (*pyknp.juman.mlist.MList method*), 16

`spec()` (*pyknp.juman.morpheme.Morpheme method*), 18

`spec()` (*pyknp.knp.blist.BList method*), 21

`spec()` (*pyknp.knp.bunsetsu.Bunsetsu method*), 22

`spec()` (*pyknp.knp.tag.Tag method*), 25

`spec()` (*pyknp.knp.tlist.TList method*), 23

`sprint_tree()` (*in module pyknp.knp.drawtree*), 27

`sprint_tree()` (*pyknp.knp.blist.BList method*), 21

`sprint_tree()` (*pyknp.knp.tlist.TList method*), 23

T

`Tag` (*class in pyknp.knp.tag*), 23

`tag_id` (*pyknp.knp.tag.Tag attribute*), 23

`tag_list()` (*pyknp.knp.blist.BList method*), 20

`tag_list()` (*pyknp.knp.bunsetsu.Bunsetsu method*), 22

`tag_list()` (*pyknp.knp.tlist.TList method*), 23

`tid` (*pyknp.knp.pas.Argument attribute*), 26

`TList` (*class in pyknp.knp.tlist*), 23

Y

`yomi` (*pyknp.juman.morpheme.Morpheme attribute*), 17