budou Documentation

Shuhei Iitsuka

Nov 08, 2022

Contents:

CHAPTER 1

budou package

1.1 Submodules

1.2 budou.budou module

Budou: an automatic organizer tool for beautiful line breaking in CJK

Usage: budou [–segmenter=<seg>] [–language=<lang>] [–separator=<separator>] [–classname=<class>] [–inlinestyle] [–wbr] [<source>] budou -h | –help budou -v | –version

Options: -h –help Show this screen.

-v –version Show version.

--segmenter=<segmenter> Segmenter to use [default: nlapi].

--language=<language> Language the source in.

--separator=<separator> Custom separator instead of SPAN tags, when used classname and inlinestyle are ignored

--classname=<classname> Class name for output SPAN tags. Use comma-separated value to specify multiple classes.

--inlinestyle Add `display:inline-block` as inline style attribute.

--wbr User WBR tag for serialization instead of inline-block SPAN tags.

`budou.budou.`**`authenticate`**(*json_path=None*)

Gets a Natural Language API parser by authenticating the API.

This method is deprecated. Please use `budou.parser.get_parser` to obtain a parser instead.

Parameters **`json_path`** (*`str, optional`*) – The file path to the service account's credentials.

Returns Parser. (`budou.parser.NLAPIParser`)

budou.budou.**main**()

Budou main method for the command line tool.

budou.budou.**parse**(*source*, *segmenter='nlapi'*, *language=None*, *max_length=None*, *classname=None*, *attributes=None*, *inlinestyle=False*, *wbr=False*, ***kwargs*)

Parses input source.

Parameters

- **source** (*str*) – Input source to process.
- **segmenter** (*str, optional*) – Segmenter to use [default: nlapi].
- **language** (*str, optional*) – Language code.
- **max_length** (*int, optional*) – Maximum length of a chunk.
- **classname** (*str, optional*) – Class name of output SPAN tags.
- **attributes** (*dict, optional*) – Attributes for output SPAN tags.
- **inlinestyle** (*bool, optional*) – Add `display:inline-block` as inline style attribute.
- **wbr** (*bool, optional*) – User WBR tag for serialization.

Returns Results in a dict. `chunks` holds a list of chunks (*budou.chunk.ChunkList*) and `html_code` holds the output HTML code.

1.3 budou.cachefactory module

Budou cache factory class.

class budou.cachefactory.**AppEngineMemcache**

Bases: *budou.cachefactory.BudouCache*

Cache system with `google.appengine.api.memcache` backend.

memcache

Memcache service.

Type `google.appengine.api.memcache`

get(*key*)

Gets a value by a key.

Parameters **key** (*str*) – Key to retrieve the value.

Returns Retrieved value (str or None).

set(*key*, *val*)

Sets a value in a key.

Parameters

- **key** (*str*) – Key for the value.
- **val** (*str*) – Value to set.

class budou.cachefactory.**BudouCache**

Bases: `object`

Base class for cache system.

get (*key*)
Abstract method: Gets a value by a key.

Parameters **key** (*str*) – Key to retrieve the value.

Returns Retrieved value (str or None).

Raises `NotImplementedError` – If it's not implemented.

set (*key*, *val*)
Abstract method: Sets a value in a key.

Parameters

- **key** (*str*) – Key for the value.
- **val** (*str*) – Value to set.

Raises `NotImplementedError` – If it's not implemented.

class budou.cachefactory.**PickleCache** (*filename*)
Bases: *budou.cachefactory.BudouCache*

Cache system with `pickle` backend.

Parameters **filename** (*str*) – The file path to the cache file.

filename
The file path to the cache file.

Type str

DEFAULT_FILE_NAME = '/tmp/budou-cache.pickle'
The default path to the cache file.

get (*key*)
Gets a value by a key.

Parameters **key** (*str*) – Key to retrieve the value.

Returns: Retrieved value (str or None).

set (*key*, *val*)
Sets a value in a key.

Parameters

- **key** (*str*) – Key for the value.
- **val** (*str*) – Value to set.

budou.cachefactory.**load_cache** (*filename=None*)
Returns a cache service.

If Google App Engine Standard Environment's memcache is available, this uses memcache as the backend. Otherwise, this uses `pickle` to cache the outputs in the local file system.

Parameters **filename** (*str, optional*) – The file path to the cache file. This is used only when `pickle` is used as the backend.

Returns A cache system (*budou.cachefactory.BudouCache*)

1.4 budou.chunk module

Chunk module as a unit of word segment with helpers.

class budou.chunk.**Chunk**(*word*, *pos=None*, *label=None*, *dependency=None*)

A unit for word segmentation.

word

Surface word of the chunk.

Type str

pos

Part of speech.

Type str, optional

label

Label information.

Type str, optional

dependency

Dependency to neighbor words. `None` for no dependency, `True` for dependency to the following word, and `False` for the dependency to the previous word.

Type bool, optional

Parameters

- **word** (*str*) – Surface word of the chunk.
- **pos** (*str, optional*) – Part of speech.
- **label** (*str, optional*) – Label information.
- **dependency** (*bool, optional*) – Dependency to neighbor words. `None` for no dependency, `True` for dependency to the following word, and `False` for the dependency to the previous word.

classmethod breakline()

Creates breakline Chunk.

Returns A chunk (*budou.chunk.Chunk*)

has_cjk()

Checks if the word of the chunk contains CJK characters.

This is using unicode codepoint ranges from https://github.com/nltk/nltk/blob/develop/nltk/tokenize/util.py#L149

Returns True if the chunk has any CJK character.

Return type bool

is_open_punct()

Whether the chunk is an open punctuation mark.

Ps: Punctuation, open (e.g. opening bracket characters) Pi: Punctuation, initial quote (e.g. opening quotation mark) See also https://en.wikipedia.org/wiki/Unicode_character_property

Returns True if it is an open punctuation mark.

Return type bool

is_punct()

Whether the chunk is a punctuation mark.

See also https://en.wikipedia.org/wiki/Unicode_character_property

Returns True if it is a punctuation mark.

Return type bool

is_space()
Whether the chunk is a space.

Returns True if it is a space.

Return type bool

serialize()
Returns serialized chunk data in dictionary.

classmethod space()
Creates space Chunk.

Returns A chunk (*budou.chunk.Chunk*)

class budou.chunk.**ChunkList**(**args*)
Bases: _abcoll.MutableSequence

List of *budou.chunk.Chunk* with some helpers.

This list accepts only instances of *budou.chunk.Chunk*.

Example

```
from budou.chunk import Chunk, ChunkList
chunks = ChunkList(Chunk('abc'), Chunk('def'))
chunks.append(Chunk('ghi'))  # OK
chunks.append('jkl')         # NG
```

Parameters **args** (list of *budou.chunk.Chunk*) – Initial values included in the list.

get_overlaps(*offset*, *length*)
Returns chunks overlapped with the given range.

Parameters

- **offset** (*int*) – Begin offset of the range.
- **length** (*int*) – Length of the range.

Returns Overlapped chunks. (*budou.chunk.ChunkList*)

html_serialize(*attributes*, *max_length=None*, *use_wbr=False*)
Returns concatenated HTML code with SPAN tag.

Parameters

- **attributes** (*dict*) – A map of name-value pairs for attributes of output SPAN tags.
- **max_length** (*int, optional*) – Maximum length of span enclosed chunk.
- **use_wbr** (*bool, optional*) – Use WBR tag to serialize the output.

Returns The organized HTML code. (str)

insert(*index*, *value*)
S.insert(index, object) – insert object before index

resolve_dependencies()
Resolves chunk dependency by concatenating them.

separator_serialize(*separator*)
Returns concatenated chunks with a custom separator in between.

Returns The organized string with custom separator (str)

span_serialize(*attributes*, *max_length=None*)
Returns concatenated HTML code with SPAN tag.

Parameters

- **attributes** (*dict*) – A map of name-value pairs for attributes of output SPAN tags.
- **max_length** (*int, optional*) – Maximum length of span enclosed chunk.

Returns The organized HTML code. (str)

swap(*old_chunks*, *new_chunk*)
Swaps old consecutive chunks with new chunk.

Parameters

- **old_chunks** (*budou.chunk.ChunkList*) – List of consecutive Chunks to be removed.
- **new_chunk** (*budou.chunk.Chunk*) – A Chunk to be inserted.

wbr_serialize()
Returns concatenated HTML code with WBR tag. This is still experimental.

Returns The organized HTML code. (str)

1.5 budou.mecabsegmenter module

MeCab based Segmenter.

Word segmenter module powered by MeCab. You need to install MeCab to use this segmenter. The easiest way to install MeCab is to run `make install-mecab`. The script will download source codes from GitHub and build the tool. It also setup IPAdic, a standard dictionary for Japanese.

class budou.mecabsegmenter.**MecabSegmenter**
Bases: *budou.segmenter.Segmenter*

MeCab Segmenter.

tagger
MeCab Tagger to parse the input sentence.

Type MeCab.Tagger

supported_languages
List of supported languages' codes.

Type list of str

segment(*source*, *language=None*)
Returns a chunk list from the given sentence.

Parameters

- **source** (*str*) – Source string to segment.
- **language** (*str, optional*) – A language code.

Returns A chunk list. (*budou.chunk.ChunkList*)

Raises `ValueError` – If `language` is given and it is not included in `supported_languages`.

`supported_languages = set(['ja'])`

1.6 budou.nlapisegmenter module

Natural Language API based Segmenter.

Word segmenter module powered by Cloud Natural Language API. You need to enable the API in your Google Cloud Platform project before you use this module.

Example

Once you enabled the API, download a service account's credentials and set as *GOOGLE_APPLICATION_CREDENTIALS* environment variable.

```
$ export GOOGLE_APPLICATION_CREDENTIALS='/path/to/credentials.json'
```

Alternatively, you can also pass the path to your credentials file to the module.

```
segmenter = budou.segmenter.NLAPISegmenter(
    credentials_path='/path/to/credentials.json')
```

This module is equipped with caching system not to make multiple requests for the same source sentence because making request to the API may incur costs. The caching system is provided by *budou.cachefactory*, and a proper caching system is chosen to be used based on the environment.

class budou.nlapisegmenter.**NLAPISegmenter**(*cache_filename*, *credentials_path*, *use_entity*, *use_cache*, *cache_discovery=True*, *service=None*)

Bases: *budou.segmenter.Segmenter*

Natural Language API Segmenter.

service

A resource object for interacting with Cloud Natural Language API.

cache_filename

File path to the cache file.

Type str

supported_languages

List of supported languages' codes.

Type list of str

Parameters

- **cache_filename** (*str, optional*) – File path to the pickle file for caching. The file is created automatically if not exist. If the environment is Google App Engine Standard Environment and memcache service is available, it is used for caching and the pickle file won't be generated.
- **credentials_path** (*str, optional*) – File path to the service account's credentials file. If no file path is specified, it tries to authenticate with default credentials.

- **use_entity** (*bool, optional*) – Whether to use entity analysis results to wrap entity names in the output.
- **use_cache** (*bool, optional*) – Whether to use a cache system.
- **cache_discovery** (*bool, optional*) – Whether to use the cache to build the natural language API service [default: True]. When using oauth2client >= 4.0.0 or google-auth, its value should be False.
- **service** (`googleapiclient.discovery.Resource`, optional) – A Resource object for interacting with Cloud Natural Language API. If this is given, the constructor skips the authentication process and use this service instead.

segment (*source*, *language=None*)

Returns a chunk list from the given sentence.

Parameters

- **source** (*str*) – Source string to segment.
- **language** (*str, optional*) – A language code.

Returns A chunk list. (*budou.chunk.ChunkList*)

Raises `ValueError` – If `language` is given and it is not included in `supported_languages`.

supported_languages = set([u'ja', u'ko', u'zh', u'zh-CN', u'zh-HK', u'zh-Hant', u'zh-T

budou.nlapisegmenter.**generate_hash** (*classname*, *funcname*, **args*, ***kwargs*)

1.7 budou.parser module

Parser modules.

Parser modules have `parse` method which processes the input text into a list of chunks and a HTML snippet.

Examples

```
import budou
parser = budou.get_parser('nlapi')
results = parser.parse('Google Home ', classname='w')
print(results['html_code'])
# <span>Google <span class="w">Home </span>
# <span class="w"></span></span>

chunks = results['chunks']
print(chunks[1].word)  # Home
```

class budou.parser.**MecabParser**

Bases: *budou.parser.Parser*

Parser built on Mecab Segmenter (*budou.mecabsegmenter.MecabSegmenter*).

segmenter

Segmenter module.

Type *budou.mecabsegmenter.MecabSegmenter*

class budou.parser.**NLAPIParser**(***options*)

Bases: *budou.parser.Parser*

Parser built on Cloud Language API Segmenter (*budou.nlapisegmenter.NLAPISegmenter*).

Parameters

- **cache_filename** (*string, optional*) – the path to the cache file.
- **credentials_path** (*string, optional*) – the path to the service account's credentials file.
- **use_entity** (*bool, optional*) – Whether to use entity analysis results to wrap entity names in the output.
- **use_cache** (*bool, optional*) – Whether to use a cache system.
- **service** (googleapiclient.discovery.Resource, optional) – A Resource object for interacting with Cloud Natural Language API. If this is given, the constructor skips the authentication process and use this service instead.

segmenter

Segmenter module.

Type *budou.nlapisegmenter.NLAPISegmenter*

class budou.parser.**Parser**

Bases: object

Abstract parser class:

segmenter

Segmenter module.

Type *budou.segmenter.Segmenter*

parse(*source*, *language=None*, *classname=None*, *max_length=None*, *attributes=None*, *inlinestyle=False*, *wbr=False*)

Parses the source sentence to output organized HTML code.

Parameters

- **source** (*str*) – Source sentence to process.
- **language** (*str, optional*) – Language code.
- **max_length** (*int, optional*) – Maximum length of a chunk.
- **attributes** (*dict, optional*) – Attributes for output SPAN tags.
- **inlinestyle** (*bool, optional*) – Add display:inline-block as inline style attribute.
- **wbr** (*bool, optional*) – User WBR tag for serialization.

Returns A dictionary containing chunks (*budou.chunk.ChunkList*) and html_code (str).

class budou.parser.**TinysegmenterParser**

Bases: *budou.parser.Parser*

Parser built on TinySegmenter Segmenter (*budou.tinysegmentersegmenter.TinysegmenterSegmenter*).

segmenter

Segmenter module.

Type *budou.tinysegmentersegmenter.TinysegmenterSegmenter*

budou.parser.**get_parser**(*segmenter*, ***options*)

Gets a parser.

Parameters

- **segmenter** (*str*) – Segmenter to use.
- **options** (*dict, optional*) – Optional settings.

Returns Parser (*budou.parser.Parser*)

Raises ValueError – If unsupported segmenter is specified.

budou.parser.**parse_attributes**(*attributes=None*, *classname=None*, *inlinestyle=False*)

Parses attributes,

Parameters

- **attributes** (*dict*) – Input attributes.
- **classname** (*str, optional*) – Class name of output SPAN tags.
- **inlinestyle** (*bool, optional*) – Add display:inline-block as inline style attribute.

Returns Parsed attributes. (dict)

budou.parser.**preprocess**(*source*)

Removes unnecessary break lines and white spaces.

Parameters **source** (*str*) – Input sentence.

Returns Preprocessed sentence. (str)

1.8 budou.segmenter module

Segmenter module.

class budou.segmenter.**Segmenter**

Bases: object

Base class for Segmenter modules.

segment(*source*, *language=None*)

Returns a chunk list from the given sentence.

Parameters

- **source** (*str*) – Source string to segment.
- **language** (*str, optional*) – A language code.

Returns A chunk list. (*budou.chunk.ChunkList*)

Raises NotImplementedError – If not implemented.

1.9 budou.tinysegmentersegmenter module

TinySegmenter based Segmenter.

Word segmenter module powered by TinySegmenter, a compact Japanese tokenizer originally developed by Taku Kudo. This is built on its Python port (https://pypi.org/project/tinysegmenter3/) developed by Tatsuro Yasukawa.

class budou.tinysegmentersegmenter.**TinysegmenterSegmenter**

Bases: *budou.segmenter.Segmenter*

TinySegmenter based Segmenter.

supported_languages

List of supported languages' codes.

Type list of str

segment(*source*, *language=None*)

Returns a chunk list from the given sentence.

Parameters

- **source** (*str*) – Source string to segment.
- **language** (*str, optional*) – A language code.

Returns A chunk list. (*budou.chunk.ChunkList*)

Raises ValueError – If language is given and it is not included in supported_languages.

supported_languages = set(['ja'])

budou.tinysegmentersegmenter.**is_hiragana**(*word*)

Checks is the word is a Japanese hiragana.

This is using the unicode codepoint range for hiragana. https://en.wikipedia.org/wiki/Hiragana_(Unicode_block)

Parameters **word** (*str*) – A word.

Returns True if the word is a hiragana.

Return type bool

1.10 Module contents

Package indicator for budou.

CHAPTER 2

Indices and tables

- genindex
- modindex
- search

Python Module Index

b

Index

R

S

T

W