

---

# **budou Documentation**

**ShuheI litsuka**

**Nov 06, 2019**



---

## Contents:

---

<b>1</b>	<b>budou package</b>	<b>1</b>
1.1	Submodules . . . . .	1
1.2	budou.budou module . . . . .	1
1.3	budou.cachefactory module . . . . .	2
1.4	budou.chunk module . . . . .	3
1.5	budou.mecabsegmenter module . . . . .	6
1.6	budou.nlapisegmenter module . . . . .	6
1.7	budou.parser module . . . . .	8
1.8	budou.segmenter module . . . . .	10
1.9	budou.tinysegmentersegmenter module . . . . .	10
1.10	Module contents . . . . .	11
<b>2</b>	<b>Indices and tables</b>	<b>13</b>
	<b>Python Module Index</b>	<b>15</b>
	<b>Index</b>	<b>17</b>



## 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>] [-classname=<class>] [-inlinestyle] <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.

**--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.

`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, \*\*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.

**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](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](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 budu.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*)

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)

**insert** (*index, value*)

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

**resolve\_dependencies** ()

Resolves chunk dependency by concatenating them.

**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.

## 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, ser-  
                                           vice=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'
```

## 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*)

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.

**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 if the word is a Japanese hiragana.

This is using the unicode codepoint range for hiragana. [https://en.wikipedia.org/wiki/Hiragana\\_\(Unicode\\_block\)](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`



### b

- `budou`, [11](#)
- `budou.budou`, [1](#)
- `budou.cachefactory`, [2](#)
- `budou.chunk`, [3](#)
- `budou.mecabsegmenter`, [6](#)
- `budou.nlapisegmenter`, [6](#)
- `budou.parser`, [8](#)
- `budou.segmenter`, [10](#)
- `budou.tinysegmentersegmenter`, [10](#)



## A

AppEngineMemcache (class in budou.cachefactory), 2

authenticate() (in module budou.budou), 1

## B

breakline() (budou.chunk.Chunk class method), 4

budou (module), 11

budou.budou (module), 1

budou.cachefactory (module), 2

budou.chunk (module), 3

budou.mecabsegmenter (module), 6

budou.nlapisegmenter (module), 6

budou.parser (module), 8

budou.segmenter (module), 10

budou.tinysegmentersegmenter (module), 10

BudouCache (class in budou.cachefactory), 2

## C

cache\_filename (budou.nlapisegmenter.NLAPISegmenter attribute), 7

Chunk (class in budou.chunk), 3

ChunkList (class in budou.chunk), 5

## D

DEFAULT\_FILE\_NAME (budou.cachefactory.PickleCache attribute), 3

dependency (budou.chunk.Chunk attribute), 4

## F

filename (budou.cachefactory.PickleCache attribute), 3

## G

get() (budou.cachefactory.AppEngineMemcache method), 2

get() (budou.cachefactory.BudouCache method), 2

get() (budou.cachefactory.PickleCache method), 3

get\_overlaps() (budou.chunk.ChunkList method), 5

get\_parser() (in module budou.parser), 9

## H

has\_cjk() (budou.chunk.Chunk method), 4

html\_serialize() (budou.chunk.ChunkList method), 5

## I

insert() (budou.chunk.ChunkList method), 5

is\_hiragana() (in module budou.tinysegmentersegmenter), 10

is\_open\_punct() (budou.chunk.Chunk method), 4

is\_punct() (budou.chunk.Chunk method), 4

is\_space() (budou.chunk.Chunk method), 4

## L

label (budou.chunk.Chunk attribute), 4

load\_cache() (in module budou.cachefactory), 3

## M

main() (in module budou.budou), 1

MecabParser (class in budou.parser), 8

MecabSegmenter (class in budou.mecabsegmenter), 6

memcache (budou.cachefactory.AppEngineMemcache attribute), 2

## N

NLAPIParser (class in budou.parser), 8

NLAPISegmenter (class in budou.nlapisegmenter), 7

## P

parse() (budou.parser.Parser method), 9

parse() (in module budou.budou), 1

parse\_attributes() (in module budou.parser), 9

Parser (class in budou.parser), 8

PickleCache (class in budou.cachefactory), 3

pos (budou.chunk.Chunk attribute), 3

`preprocess()` (in module *budou.parser*), 9

## R

`resolve_dependencies()` (*budou.chunk.ChunkList* method), 5

## S

`segment()` (*budou.mecabsegmenter.MecabSegmenter* method), 6  
`segment()` (*budou.nlapisegmenter.NLAPISegmenter* method), 7  
`segment()` (*budou.segmenter.Segmenter* method), 10  
`segment()` (*budou.tinysegmentersegmenter.TinysegmenterSegmenter* method), 10  
`segmenter` (*budou.parser.MecabParser* attribute), 8  
`segmenter` (*budou.parser.NLAPIParser* attribute), 8  
`segmenter` (*budou.parser.Parser* attribute), 8  
`segmenter` (*budou.parser.TinysegmenterParser* attribute), 9  
`Segmenter` (class in *budou.segmenter*), 10  
`serialize()` (*budou.chunk.Chunk* method), 5  
`service` (*budou.nlapisegmenter.NLAPISegmenter* attribute), 7  
`set()` (*budou.cachefactory.AppEngineMemcache* method), 2  
`set()` (*budou.cachefactory.BudouCache* method), 2  
`set()` (*budou.cachefactory.PickleCache* method), 3  
`space()` (*budou.chunk.Chunk* class method), 5  
`supported_languages` (*budou.mecabsegmenter.MecabSegmenter* attribute), 6  
`supported_languages` (*budou.nlapisegmenter.NLAPISegmenter* attribute), 7  
`supported_languages` (*budou.tinysegmentersegmenter.TinysegmenterSegmenter* attribute), 10  
`swap()` (*budou.chunk.ChunkList* method), 5

## T

`tagger` (*budou.mecabsegmenter.MecabSegmenter* attribute), 6  
`TinysegmenterParser` (class in *budou.parser*), 9  
`TinysegmenterSegmenter` (class in *budou.tinysegmentersegmenter*), 10

## W

`word` (*budou.chunk.Chunk* attribute), 3