bosonnlp.py Documentation

Release 0.3.0

BosonData

August 25, 2014

Contents

BosonNLP HTTP API SDK

CHAPTER 1

bosonnlp GitHub PyPI *pip* :

```
$ pip install bosonnlp
```

bosonnlp MIT

```
>>> from bosonnlp import BosonNLP
>>> nlp = BosonNLP('YOUR_API_TOKEN')
>>> nlp.sentiment('')
[[0.4464756252294154, 0.5535243747705846]]
```

BosonNLP BosonNLP HTTP API

CHAPTER 3

API

class bosonnlp.**BosonNLP** (*token*, *bosonnlp_url='http://api.bosonnlp.com'*)

BosonNLP HTTP API

Parameters

- **token** (*string*) – API API Token
- **bosonnlp_url** (*string*) – BosonNLP HTTP API URL *http://api.bosonnlp.com*

sentiment (*contents*, *news=False*)

BosonNLP

Parameters

- **contents** (*string or sequence of string*) –
- **news** (*bool*) – False

Returns

Raises HTTPError API

```
>>> nlp = BosonNLP('YOUR_API_TOKEN')
>>> nlp.sentiment('')
[[0.4464756252294154, 0.5535243747705846]]
>>> nlp.sentiment(['', ''])
[[0.4464756252294154, 0.5535243747705846], [0.6739600397344988, 0.3260399602655012]]
```

convert_time (*content*, *basetime=None*)

BosonNLP

Parameters

- **content** (*string*) –
- **basetime** (*int or datetime.datetime*) – datetime

Raises HTTPError API

Returns

```
>>> nlp = BosonNLP('YOUR_API_TOKEN')
>>> nlp.convert_time("2013")
{u'timestamp': u'2013-02-28 16:30:29'}
>>> import datetime
>>> nlp.convert_time("83", datetime.datetime.today())
{u'timespan': [u'2014-08-25 20:00:00', u'2014-08-26 15:00:00']}
```

classify(*contents*)
BosonNLP

Parameters **contents** (*string or sequence of string*) –

Returns

Raises HTTPError API

```
>>> nlp = BosonNLP('YOUR_API_TOKEN')
>>> nlp.classify('')
[5]
>>> nlp.classify(['',
...               '',
...               'Facebook'])
[5, 4, 8]
```

suggest(*word*, *top_k=None*)
BosonNLP

Parameters

- **word** (*string*) –
- **top_k** (*int*) – 10 100

Returns

Raises HTTPError API

```
>>> nlp = BosonNLP('YOUR_API_TOKEN')
>>> nlp.suggest('python', top_k=1)
[[0.9999999999999992, 'python/x']]
```

extract_keywords(*text*, *top_k=None*, *segmented=False*)
BosonNLP

Parameters

- **text** (*string*) –
- **top_k** (*int*) – 100
- **segmented** (*bool*) – Falsetext True

Returns

Raises HTTPError API

```
>>> nlp = BosonNLP('YOUR_API_TOKEN')
>>> nlp.extract_keywords('', top_k=2)
[[0.4580507649282757, ''], [0.44467176143180404, '']]
```

depparser(*contents*)
BosonNLP

Parameters **contents** (*string or sequence of string*) –

Returns

Raises HTTPError API

```
>>> nlp = BosonNLP('YOUR_API_TOKEN')
>>> nlp.depparser('')
[{'role': ['SBJ', 'ROOT', 'NMOD', 'VMOD'],
  'head': [1, -1, 3, 1],
```

```
    'word': ['', '', '', ''],
    'tag': ['PN', 'VC', 'M', 'NN']}]
>>> nlp.depparser(['', ''])
[{'role': ['SBJ', 'ROOT', 'NMOD', 'VMOD'],
  'head': [1, -1, 3, 1],
  'word': ['', '', '', ''],
  'tag': ['PN', 'VC', 'M', 'NN']},
 {'role': ['DEC', 'NMOD', 'ROOT'],
  'head': [1, 2, -1],
  'word': ['', '', ''],
  'tag': ['VA', 'DEC', 'NN']}]
```

ner (*contents*, *sensitivity=None*)
BosonNLP

Parameters

- **contents** (*string or sequence of string*) –
- **sensitivity** (*int 3*) – 1 5

Returns

Raises HTTPError API

```
>>> nlp = BosonNLP('YOUR_API_TOKEN')
>>> nlp.ner(' ')
[{'entity': [[0, 2, 'product_name'], [3, 4, 'person_name']],
  'tag': ['ns', 'n', 'n', 'nr'],
  'word': ['', '', '', '']}]
>>> nlp.ner([' ', 'XP'])
[{'entity': [[0, 2, 'product_name'], [3, 4, 'person_name']],
  'tag': ['ns', 'n', 'n', 'nr'],
  'word': ['', '', '', '']},
 {'entity': [[0, 2, 'product_name'], [3, 4, 'time']],
  'tag': ['nt', 'x', 'nl', 't', 'ad', 'v'],
  'word': ['', 'XP', '', '', '', '']}]
```

tag (*contents*)
BosonNLP

Parameters **contents** (*string or sequence of string*) –

Returns

Raises HTTPError API

```
>>> nlp = BosonNLP('YOUR_API_TOKEN')
>>> nlp.tag(' ')
[{'tag': ['NR', 'NN', 'NN', 'NR'],
  'word': ['', '', '', '']}]
>>> nlp.tag([' ', 'XP'])
[{'tag': ['NR', 'NN', 'NN', 'NR'],
  'word': ['', '', '', '']},
 {'tag': ['NR', 'NN', 'NN', 'NN', 'NT', 'AD', 'VV'],
  'word': ['', 'XP', '', '', '', '', '']}]
```

cluster (*contents*, *task_id=None*, *alpha=None*, *beta=None*, *timeout=1800*)
BosonNLP

Parameters

- **contents** (*sequence of string or sequence of (_id, text) or sequence of {'_id': _id, 'text': text}*) – (_id, text) {'_id': _id, 'text': text} _id _id
- **task_id** (*string*) – task_id
- **alpha** (*float*) – 0.8 cluster
- **beta** (*float*) – 0.45 cluster
- **timeout** (*float*) – 1800 30

Returns

Raises `HTTPError` - API

`TaskError` -

`TimeoutError` - *timeout*

```
>>> nlp = BosonNLP('YOUR_API_TOKEN')
>>> nlp.cluster(['', '', '', '',
...              '', '', ''])
[{'_id': 0, 'list': [0, 1], 'num': 2}]
```

create_cluster_task(*contents=None*, *task_id=None*)
`ClusterTask`

Parameters

- **contents** (`None` or sequence of string or sequence of (_id, text) or sequence of {'_id': _id, 'text': text}) – (_id, text) {'_id': _id, 'text': text} _id _id `None`
- **task_id** (*string*) – `None` task_id

Raises

`HTTPError` - **API** *contents* `None` `push`

Returns `ClusterTask`

comments(*contents*, *task_id=None*, *alpha=None*, *beta=None*, *timeout=1800*)
BosonNLP

Parameters

- **contents** (*sequence of string or sequence of (_id, text) or sequence of {'_id': _id, 'text': text}*) – (_id, text) {'_id': _id, 'text': text} _id _id
- **task_id** (*string*) – `None` task_id
- **alpha** (*float*) – 0.8 cluster
- **beta** (*float*) – 0.45 cluster
- **timeout** (*float*) – 1800 30

Returns

Raises `HTTPError` - API

`TaskError` -

`TimeoutError` - *timeout*

```
>>> nlp = BosonNLP('YOUR_API_TOKEN')
>>> nlp.comments(['', '', '', '',
...               '', '', ''] * 2)
[{'_id': 0, 'list': [['', 3], ['', 10]],
```

```
 'num': 2, 'opinion': ''},
{'_id': 1, 'list': [['', 4], ['', 11]],
 'num': 2, 'opinion': ''},
{'_id': 2, 'list': [['', 5], ['', 12]],
 'num': 2, 'opinion': ''},
{'_id': 3, 'list': [['', 6], ['', 13]],
 'num': 2, 'opinion': ''}]
```

create_comments_task(*contents=None*, *task_id=None*)

CommentsTask

Parameters

- **contents** (None or sequence of string or sequence of (_id, text) or sequence of {'_id': _id, 'text': text}) – (_id, text) {'_id': _id, 'text': text} _id _id None
- **task_id** (*string*) – None task_id

Raises

HTTPError - API *contents* None push

Returns CommentsTask

class bosonnlp.**ClusterTask**(*nlp*, *contents=None*, *task_id=None*)

create_cluster_task

```
>>> nlp = BosonNLP('YOUR_API_TOKEN')
>>> cluster = nlp.create_cluster_task()
```

push

```
>>> cluster.push(['', '', '', ''])
>>> cluster.push([''])
>>> cluster.push(['', ''])
```

analysis

```
>>> cluster.analysis()
```

wait_until_complete

```
>>> cluster.wait_until_complete(2 * 60)  #  2
```

status

```
>>> cluster.status()
'done'
```

result

```
>>> cluster.result()
[{'_id': '9e90c56e-f1bb-4605-b995-304af733207a',
  'list': ['9e90c56e-f1bb-4605-b995-304af733207a',
           'a3feff6b-6d1b-4f46-a2d8-0eea25c7f17f'],
  'num': 2}]
```

clear

```
>>> cluster.clear()
True
```

Parameters **nlp** – BosonNLP cluster

analysis (*alpha=None*, *beta=None*)

Parameters

- **alpha** (*float*) – 0.8 cluster
- **beta** (*float*) – 0.45 cluster

Raises HTTPError - API

clear ()

Returns

Raises HTTPError - API

push (*contents*)

Parameters

contents (*sequence of string or sequence of (_id, text) or sequence of {'_id': _id, 'text': text}*) – (_id, text) {'_id': _id, 'text': text} _id _id

Raises HTTPError - API

result ()

Returns

Raises HTTPError - API

status ()

Returns

received running done

Raises HTTPError - API

TaskNotFoundError -

TaskError -

wait_until_complete (*timeout=None*)

Parameters

timeout (*float*) – None

Raises HTTPError - API

TaskNotFoundError -

TaskError -

TimeoutError - *timeout*

class bosonnlp.**CommentsTask** (*nlp*, *contents=None*, *task_id=None*)

create_comments_task

```
>>> nlp = BosonNLP('YOUR_API_TOKEN')
>>> comments = nlp.create_comments_task()
```

push

```
>>> comments.push(['', '', '', ''] * 2)
>>> comments.push([''] * 2)
>>> comments.push(['', ''] * 2)
```

analysis

```
>>> comments.analysis()
```

wait_until_complete

```
>>> comments.wait_until_complete(2 * 60)  #  2
```

status

```
>>> comments.status()
'done'
```

result

```
>>> comments.result()
[{'_id': 0,
  'list': [['', '19c248e3-605b-4785-8785-ccd2d1b034cc'],
           ['', '576d1d08-ff02-4bc5-9edf-fbc7a6915e44']],
  'num': 2,
  'opinion': ''},
 {'_id': 2,
  'list': [['', 'a75e24bd-8597-4865-8254-2e9cab229770'],
           ['', '47db4d92-6328-45cd-98f8-099691e82c07']],
  'num': 2,
  'opinion': ''},
 {'_id': 4,
  'list': [['', 'da0cd13f-4f13-4476-a541-6214df3b4dd9'],
           ['', '89aecf45-4b78-4522-9ed2-ea76ed552f24']],
  'num': 2,
  'opinion': ''},
 {'_id': 5,
  'list': [['', 'a5c1f6a9-b6a6-4877-b073-0b59bc67fa48'],
           ['', '9d38ecab-5860-44e9-9e3c-9ad4d4ed3547']],
  'num': 2,
  'opinion': ''}]
```

clear

```
>>> comments.clear()
True
```

Parameters **nlp** – BosonNLP comments

analysis(*alpha=None*, *beta=None*) **Parameters**

- **alpha** (*float*) – 0.8 cluster
- **beta** (*float*) – 0.45 cluster

Raises HTTPError - API

clear() **Returns**

Raises HTTPError - API

push(*contents*) **Parameters**

contents (*sequence of string or sequence of (_id, text) or sequence of {'_id': _id, 'text': text}*) – (_id, text) {'_id': _id, 'text': text} _id _id

Raises HTTPError - API

result() **Returns**

Raises HTTPError - API

status()

Returns

received running done

Raises HTTPError - API

TaskNotFoundError -

TaskError -

wait_until_complete(*timeout=None*)

Parameters

timeout (*float*) – None

Raises HTTPError - API

TaskNotFoundError -

TaskError -

TimeoutError - *timeout*

CHAPTER 4

Exceptions

exception bosonnlp.**HTTPError**(**args*, ***kwargs*)
An HTTP error occurred.

exception bosonnlp.**TaskNotFoundError**(**args*, ***kwargs*)

exception bosonnlp.**TaskError**(**args*, ***kwargs*)

exception bosonnlp.**TimeoutError**

b