

---

**konoha**

**himkt**

**Jan 07, 2023**



# CONTENTS

<b>1 Quick Start</b>	<b>1</b>
<b>2 Installation</b>	<b>3</b>
<b>3 API Reference</b>	<b>5</b>
3.1 Word Level Tokenizer Interface . . . . .	5
3.2 Sentence Level Tokenizer Interface . . . . .	5
3.3 Word Tokenizer Implementations . . . . .	5
3.4 Token . . . . .	7
3.5 Data classes . . . . .	7
3.6 Server . . . . .	7
<b>4 Indices and tables</b>	<b>9</b>
<b>Python Module Index</b>	<b>11</b>
<b>Index</b>	<b>13</b>



## QUICK START

Welcome to Konoha! Konoha is a library for text processing in Japanese. In Japanese, we have to split a sentence into a sequence of words, called `tokenization`. There are many tools available for tokenizing sentences, and usages of them are not the same.

Konoha provides a unified interface to use these tools. You can try Konoha only by running `docker run` on your computer.



## INSTALLATION

Konoha supports Python 3.5 or newer.

We recommend to install via pip:

```
$ pip install konoha[all]
```

If you want to install Konoha with AllenNLP integration, please run:

```
$ pip install konoha[all_with_integrations]
```

You can also install Konoha with specific tokenizer, please run:

```
$ pip install konoha[janome,kytea,mecab,sentencepiece,sudachi,nagisa] # specify one or  
↪ more of them
```

If you run *pip install konoha*, Konoha will be installed only with sentence splitter.

You can also install the development version of Konoha from the main branch of Git repository:

```
$ pip install git+https://github.com/himkt/konoha.git
```





## API REFERENCE

### 3.1 Word Level Tokenizer Interface

```
class konoha.word_tokenizer.WordTokenizer(tokenizer: str = 'MeCab', user_dictionary_path:  
Optional[str] = None, system_dictionary_path: Optional[str]  
= None, model_path: Optional[str] = None, mode:  
Optional[str] = None, dictionary_format: Optional[str] =  
None, endpoint: Optional[str] = None, ssl: Optional[bool] =  
None, port: Optional[int] = None)
```

```
batch_tokenize(texts: List[str]) → List[List[Token]]
```

Tokenize input texts

```
tokenize(text: str) → List[Token]
```

Tokenize input text

### 3.2 Sentence Level Tokenizer Interface

```
class konoha.sentence_tokenizer.SentenceTokenizer(period: Optional[str] = None, patterns:  
Optional[List[Pattern[str]]] = None)
```

### 3.3 Word Tokenizer Implementations

#### 3.3.1 Base Word Tokenizer

```
class konoha.word_tokenizers.tokenizer.BaseTokenizer(name: str)
```

Base class for word level konoha.tokenizer

```
property name: str
```

Return name of konoha.tokenizer

```
abstract tokenize(text: str) → List[Token]
```

Abstract method for konoha.tokenization

### 3.3.2 Character Tokenizer

```
class konoha.word_tokenizers.character_tokenizer.CharacterTokenizer
```

```
    tokenize(text: str)
```

```
        Abstract method for konoha.tokenization
```

### 3.3.3 MeCab Tokenizer

```
class konoha.word_tokenizers.mecab_tokenizer.MeCabTokenizer(user_dictionary_path: Optional[str]
    = None, system_dictionary_path:
    Optional[str] = None,
    dictionary_format: Optional[str] =
    None)
```

```
    tokenize(text: str) → List[Token]
```

```
        Abstract method for konoha.tokenization
```

### 3.3.4 KyTea Tokenizer

```
class konoha.word_tokenizers.kytea_tokenizer.KyTeaTokenizer(model_path: Optional[str] = None)
```

```
    tokenize(text: str) → List[Token]
```

```
        Abstract method for konoha.tokenization
```

### 3.3.5 Sentencepiece Tokenizer

```
class konoha.word_tokenizers.sentencepiece_tokenizer.SentencepieceTokenizer(model_path: str)
```

```
    tokenize(text: str) → List[Token]
```

```
        Abstract method for konoha.tokenization
```

### 3.3.6 Sudachi Tokenizer

```
class konoha.word_tokenizers.sudachi_tokenizer.SudachiTokenizer(mode: str)
```

```
    tokenize(text: str) → List[Token]
```

```
        Abstract method for konoha.tokenization
```

### 3.3.7 Janome Tokenizer

```
class konoha.word_tokenizers.janome_tokenizer.JanomeTokenizer(user_dictionary_path:
    Optional[str] = None)
```

```
    tokenize(text: str) → List[Token]
```

```
        Abstract method for konoha.tokenization
```

### 3.3.8 nagisa Tokenizer

**class** `konoha.word_tokenizers.nagisa_tokenizer.NagisaTokenizer`

**tokenize**(*text: str*) → List[*Token*]

Abstract method for `konoha.tokenization`

### 3.3.9 Whitespace Tokenizer

**class** `konoha.word_tokenizers.whitespace_tokenizer.WhitespaceTokenizer`

Simple rule-based word tokenizer.

**tokenize**(*text: str*) → List[*Token*]

Abstract method for `konoha.tokenization`

## 3.4 Token

(Deprecated)

## 3.5 Data classes

### 3.5.1 Token

**class** `konoha.data.token.Token`(*surface: str, postag: Optional[str] = None, postag2: Optional[str] = None, postag3: Optional[str] = None, postag4: Optional[str] = None, inflection: Optional[str] = None, conjugation: Optional[str] = None, base\_form: Optional[str] = None, yomi: Optional[str] = None, pron: Optional[str] = None, normalized\_form: Optional[str] = None*)

Token class for `konoha`.

### 3.5.2 Resource

**class** `konoha.data.resource.Resource`(*path: Optional[str]*)

**download\_from\_s3**(*path: str*) → str

Download file(s) from Amazon S3.

## 3.6 Server

TBD



## INDICES AND TABLES

- genindex
- modindex
- search



## PYTHON MODULE INDEX

### k

- `konoha.data`, 7
- `konoha.data.resource`, 7
- `konoha.data.token`, 7
- `konoha.konoha_token`, 7
- `konoha.sentence_tokenizer`, 5
- `konoha.word_tokenizer`, 5
- `konoha.word_tokenizers`, 5
- `konoha.word_tokenizers.character_tokenizer`, 5
- `konoha.word_tokenizers.janome_tokenizer`, 6
- `konoha.word_tokenizers.kytea_tokenizer`, 6
- `konoha.word_tokenizers.mecab_tokenizer`, 6
- `konoha.word_tokenizers.nagisa_tokenizer`, 6
- `konoha.word_tokenizers.sentencepiece_tokenizer`,  
6
- `konoha.word_tokenizers.sudachi_tokenizer`, 6
- `konoha.word_tokenizers.tokenizer`, 5
- `konoha.word_tokenizers.whitespace_tokenizer`,  
7





## INDEX

### B

BaseTokenizer (class in *konoha.word\_tokenizers.tokenizer*), 5  
batch\_tokenize() (*konoha.word\_tokenizer.WordTokenizer* method), 5

### C

CharacterTokenizer (class in *konoha.word\_tokenizers.character\_tokenizer*), 6

### D

download\_from\_s3() (*konoha.data.resource.Resource* method), 7

### J

JanomeTokenizer (class in *konoha.word\_tokenizers.janome\_tokenizer*), 6

### K

*konoha.data* module, 7  
*konoha.data.resource* module, 7  
*konoha.data.token* module, 7  
*konoha.konoha\_token* module, 7  
*konoha.sentence\_tokenizer* module, 5  
*konoha.word\_tokenizer* module, 5  
*konoha.word\_tokenizers* module, 5  
*konoha.word\_tokenizers.character\_tokenizer* module, 5  
*konoha.word\_tokenizers.janome\_tokenizer* module, 6  
*konoha.word\_tokenizers.kytea\_tokenizer* module, 6  
*konoha.word\_tokenizers.mecab\_tokenizer* module, 6

*konoha.word\_tokenizers.nagisa\_tokenizer* module, 6  
*konoha.word\_tokenizers.sentencepiece\_tokenizer* module, 6  
*konoha.word\_tokenizers.sudachi\_tokenizer* module, 6  
*konoha.word\_tokenizers.tokenizer* module, 5  
*konoha.word\_tokenizers.whitespace\_tokenizer* module, 7  
KyTeaTokenizer (class in *konoha.word\_tokenizers.kytea\_tokenizer*), 6

### M

MeCabTokenizer (class in *konoha.word\_tokenizers.mecab\_tokenizer*), 6  
module  
*konoha.data*, 7  
*konoha.data.resource*, 7  
*konoha.data.token*, 7  
*konoha.konoha\_token*, 7  
*konoha.sentence\_tokenizer*, 5  
*konoha.word\_tokenizer*, 5  
*konoha.word\_tokenizers*, 5  
*konoha.word\_tokenizers.character\_tokenizer*, 5  
*konoha.word\_tokenizers.janome\_tokenizer*, 6  
*konoha.word\_tokenizers.kytea\_tokenizer*, 6  
*konoha.word\_tokenizers.mecab\_tokenizer*, 6  
*konoha.word\_tokenizers.nagisa\_tokenizer*, 6  
*konoha.word\_tokenizers.sentencepiece\_tokenizer*, 6  
*konoha.word\_tokenizers.sudachi\_tokenizer*, 6  
*konoha.word\_tokenizers.tokenizer*, 5  
*konoha.word\_tokenizers.whitespace\_tokenizer*, 7

**N**

**NagisaTokenizer** (class in *konoha.word\_tokenizers.nagisa\_tokenizer*), 7  
**name** (*konoha.word\_tokenizers.tokenizer.BaseTokenizer* property), 5

**R**

**Resource** (class in *konoha.data.resource*), 7

**S**

**SentencepieceTokenizer** (class in *konoha.word\_tokenizers.sentencepiece\_tokenizer*), 6  
**SentenceTokenizer** (class in *konoha.sentence\_tokenizer*), 5  
**SudachiTokenizer** (class in *konoha.word\_tokenizers.sudachi\_tokenizer*), 6

**T**

**Token** (class in *konoha.data.token*), 7  
**tokenize()** (*konoha.word\_tokenizer.WordTokenizer* method), 5  
**tokenize()** (*konoha.word\_tokenizers.character\_tokenizer.CharacterTokenizer* method), 6  
**tokenize()** (*konoha.word\_tokenizers.janome\_tokenizer.JanomeTokenizer* method), 6  
**tokenize()** (*konoha.word\_tokenizers.kytea\_tokenizer.KyTeaTokenizer* method), 6  
**tokenize()** (*konoha.word\_tokenizers.mecab\_tokenizer.MeCabTokenizer* method), 6  
**tokenize()** (*konoha.word\_tokenizers.nagisa\_tokenizer.NagisaTokenizer* method), 7  
**tokenize()** (*konoha.word\_tokenizers.sentencepiece\_tokenizer.SentencepieceTokenizer* method), 6  
**tokenize()** (*konoha.word\_tokenizers.sudachi\_tokenizer.SudachiTokenizer* method), 6  
**tokenize()** (*konoha.word\_tokenizers.tokenizer.BaseTokenizer* method), 5  
**tokenize()** (*konoha.word\_tokenizers.whitespace\_tokenizer.WhitespaceTokenizer* method), 7

**W**

**WhitespaceTokenizer** (class in *konoha.word\_tokenizers.whitespace\_tokenizer*), 7  
**WordTokenizer** (class in *konoha.word\_tokenizer*), 5