User manual PyThaiNLP 1.4

API

Thai segment

```
from pythainlp.tokenize import word_tokenize
word_tokenize(text,engine)
```

text refers to an input text string in Thai.

engine refers to a thai word segmentation system; There are 6 systems to choose from.

1. icu (default) - pyicu has a very poor performance.
2. dict - dictionary-based tokenizer. It returns False if the message can not be wrapped.
3. mm - Maximum Matching algorithm for Thai word segmentation.
4. newmm - Maximum Matching algorithm for Thai word segmatation. Developed by Korakot Chaovavanich (https://www.facebook.com/groups/408004796247683/permalink/431283740586455/)
5. pylexto - LexTo.
6. deepcut - Deep Learning based Thai word segmentation (https://github.com/rkcosmos/deepcut)

Output: "list" ex. ['แมว','กิน']

Example

```
from pythainlp.tokenize import word_tokenize
text='ผมรักคุณนะครับโอเคบ่พวกเราเป็ นคนไทยรักภาษาไทยภาษาบ้านเกิด '
a=word_tokenize(text,engine='icu') # ['ผม', 'รัก', 'คุณ', 'นะ', 'ครับ', 'โอ', 'เค', 'บ่',
'พวก', 'เรา', 'เป็น', 'คน', 'ไทย', 'รัก', 'ภาษา', 'ไทย', 'ภาษา', 'บ้าน', 'เกิด']
b=word_tokenize(text,engine='dict') # ['ผม', 'รัก', 'คุณ', 'นะ', 'ครับ', 'โอเค', 'บ่',
'พวกเรา', 'เป็น', 'คนไทย', 'รัก', 'ภาษาไทย', 'ภาษา', 'บ้านเกิด']
c=word_tokenize(text,engine='mm') # ['ผม', 'รัก', 'คุณ', 'นะ', 'ครับ', 'โอเค', 'บ่', 'พวก
เรา', 'เป็น', 'คนไทย', 'รัก', 'ภาษาไทย', 'ภาษา', 'บ้านเกิด']
d=word_tokenize(text,engine='pylexto') # ['ผม', 'รัก', 'คุณ', 'นะ', 'ครับ', 'โอเค', 'บ่',
'พวกเรา', 'เป็น', 'คนไทย', 'รัก', 'ภาษาไทย', 'ภาษา', 'บ้านเกิด']
e=word_tokenize(text,engine='newmm') # ['ผม', 'รัก', 'คุณ', 'นะ', 'ครับ', 'โอเค', 'บ่',
'พวกเรา', 'เป็น', 'คนไทย', 'รัก', 'ภาษาไทย', 'ภาษา', 'บ้านเกิด']
```

Thai postaggers

```
from pythainlp.tag import pos_tag
pos_tag(list,engine='old')
```

engine

1. old is the UnigramTagger (default)
2. artagger is the RDR POS Tagger.

Thai romanization

```
from pythainlp.romanization import romanization
romanization(str,engine='pyicu')
```

There are 2 engines

- pyicu
- royin

data :

input "str"

returns "str"

Example

```
from pythainlp.romanization import romanization
romanization("แมว") # 'mæw'
```

Spell Check

Before using this module, please install hunspell and hunspell-th.

```
from pythainlp.spell import *
a=spell("สี่เหลียม")
print(a) # ['สี่เหลี่ยม', 'เสียเหลี่ยม', 'เหลี่ยม']
```

pythainlp.number

```
from pythainlp.number import *
```

- nttn(str) - convert thai numbers to numbers.
- nttt(str) - Thai Numbers to text.
- ntnt(str) - numbers to thai numbers.
- ntt(str) - numbers to text.
- ttn(str) - text to numbers.
- numtowords(float) - Read thai numbers (Baht) input "float" returns 'str'

Sort Thai text into List

```
from pythainlp.collation import collation
print(collation(['ไก่','ไข่','ก','ฮา'])) # ['ก', 'ไก่', 'ไข่', 'ฮา']
```

input list

returns list

Get current time in Thai

```
from pythainlp.date import now
now() # '30 พฤษภาคม 2560 18:45:24'
```

Thai WordNet

import

```
from pythainlp.corpus import wordnet
```

Use

It's like nltk.

- wordnet.synsets(word)
- wordnet.synset(name_synsets)
- wordnet.all_lemma_names(pos=None, lang="tha")
- wordnet.all_synsets(pos=None)
- wordnet.langs()
- wordnet.lemmas(word,pos=None,lang="tha")
- wordnet.lemma(name_synsets)
- wordnet.lemma_from_key(key)
- wordnet.path_similarity(synsets1,synsets2)

- wordnet.lch_similarity(synsets1,synsets2)
- wordnet.wup_similarity(synsets1,synsets2)
- wordnet.morphy(form, pos=None)
- wordnet.custom_lemmas(tab_file, lang)

Example

```
>>> from pythainlp.corpus import wordnet
>>> print(wordnet.synsets('หนึ่ง'))
[Synset('one.s.05'), Synset('one.s.04'), Synset('one.s.01'), Synset('one.n.01')]
>>> print(wordnet.synsets('หนึ่ง')[0].lemma_names('tha'))
[]
>>> print(wordnet.synset('one.s.05'))
Synset('one.s.05')
>>> print(wordnet.synset('spy.n.01').lemmas())
[Lemma('spy.n.01.spy'), Lemma('spy.n.01.undercover_agent')]
>>> print(wordnet.synset('spy.n.01').lemma_names('tha'))
['สปาย', 'สายลับ']
```

Find the most fr equent wor ds.

```
from pythainlp.rank import rank
rank(list)
```

returns dict

Example

```
>>> rank(['แมง','แมง','คน'])
Counter({'แมง': 2, 'คน': 1})
```

Incorr ect input language corr ection

```
from pythainlp.change import *
```

- texttothai(str) - eng to thai.
- texttoeng(str) - thai to eng.

Thai Char acter Clusters (T CC)

TCC : Mr.Jakkrit TeCho

grammar : Wittawat Jitkrittum (https://github.com/wittawatj/jtcc/blob/master/TCC.g)

Code : Korakot Chaovavanich

Example

```
>>> from pythainlp.tokenize import tcc
>>> tcc.tcc('ประเทศไทย')
'ป/ระ/เท/ศ/ไท/ย'
```

Enhanced Thai Char acter Cluster (ET CC)

Example

```
>>> from pythainlp.tokenize import etcc
>>> etcc.etcc('คืนความสุข')
'/คืน/ความสุข'
```

Thai Soundex

credit Korakot Chaovavanich (from https://gist.github.com/korakot/0b772e09340cac2f493868da035597e8)

- LK82
- Udom83

Example

```
>>> from pythainlp.soundex import LK82
>>> print(LK82('รถ'))
ร3000
>>> print(LK82('รด'))
ร3000
>>> print(LK82('จัน'))
จ4000
>>> print(LK82('จันทร์'))
จ4000
>>> print(Udom83('รถ'))
ร800000
```

Thai meta sound

```
Snae & Brückner. (2009). Novel Phonetic Name Matching Algorithm with a Statistical
Ontology for Analysing Names Given in Accordance with Thai Astrology. Retrieved from
https://pdfs.semanticscholar.org/3983/963e87ddc6dfdbb291099aa3927a0e3e4ea6.pdf
```

Example

```
>>> from pythainlp.MetaSound import *
>>> MetaSound('คน')
'15'
```

Thai sentiment analysis

using data from https://github.com/wannaphongcom/lexicon-thai/tree/master/ข้อความ/

```
from pythainlp.sentiment import sentiment
sentiment(str)
```

input str returns pos , neg or neutral

Util

using

```
from pythainlp.util import *
```

ngrams

for building ngrams

```
ngrams(token,num)
```

- token - list
- num - ngrams

Corpus

Thai stopword

```
from pythainlp.corpus import stopwords
stopwords = stopwords.words('thai')
```

Thai country name

```
from pythainlp.corpus import country
country.get_data()
```

Tone in Thai

```
from pythainlp.corpus import tone
tone.get_data()
```

Consonant in thai

```
from pythainlp.corpus import alphabet
alphabet.get_data()
```

Word list in thai

```
from pythainlp.corpus.thaiword import get_data # old data
get_data()
from pythainlp.corpus.newthaiword import get_data # new data
get_data()
```