คู่มือการใช้งาน PyThaiNLP 1.4

Natural language processing หรือ การประมวลภาษาธรรมชาติ โมดูล PyThaiNLP เป็นโมดูลที่ถูกพัฒนาขึ้นเพื่อพัฒนาการประมวลภาษาธรรมชาติภาษาไทยในภาษา Python และ**มันฟรี (ตลอดไป) เพื่อคนไทยและชาวโลกทุกคน !**

> เพราะโลกขับเคลื่อนต่อไปด้วยการแบ่งปัน

รองรับเฉพาะ Python 3.4 ขึ้นไปเท่านั้น

ติดตั้งใช้คำสั่ง

```
pip install pythainlp
```

วิธีติดตั้งสำหรับ Windows

ให้ทำการติดตั้ง pyicu โดยใช้ไฟล์ .whl จาก http://www.lfd.uci.edu/~gohlke/pythonlibs/#pyicu

หากใช้ python 3.5 64 bit ให้โหลด PyICU-1.9.7-cp35-cp35m-win_amd64.whl แล้วเปิด cmd ใช้คำสั่ง

```
pip install PyICU-1.9.7-cp35-cp35m-win_amd64.whl
```

แล้วจึงใช้

```
pip install pythainlp
```

ติดตั้งบน Mac

```
$ brew install icu4c --force
$ brew link --force icu4c
$ CFLAGS=-I/usr/local/opt/icu4c/include LDFLAGS=-L/usr/local/opt/icu4c/lib pip install
pythainlp
```

ข้อมูลเพิ่มเติม คลิกที่นี้

API

ตัดคำไทย

สำหรับการตัดคำไทยนั้น ใช้ API ดังต่อไปนี้

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

text คือ ข้อความในรูปแบบสตริง str เท่านั้น

engine คือ ระบบตัดคำไทย ปัจจุบันนี้ PyThaiNLP ได้พัฒนามี 6 engine ให้ใช้งานกันดังนี้

1. icu - engine ตัวดั้งเดิมของ PyThaiNLP (ความแม่นยำต่ำ) และเป็นค่าเริ่มต้น
2. dict - เป็นการตัดคำโดยใช้พจานุกรมจาก thaiword.txt ใน corpus (ความแม่นยำปานกลาง) จะคืนค่า False หากข้อความนั้นไม่สามารถตัดคำได้
3. mm - ใช้ Maximum Matching algorithm ในการตัดคำภาษาไทย - API ชุดเก่า
4. newmm - ใช้ Maximum Matching algorithm ในการตัดคำภาษาไทย โค้ดชุดใหม่ โดยใช้โค้ดคุณ Korakot Chaovavanich จาก https://www.facebook.com/groups/408004796247683/permalink/431283740586455/ มาพัฒนาต่อ
5. pylexto ใช้ LexTo ในการตัดคำ
6. deepcut ใช้ deepcut จาก https://github.com/rkcosmos/deepcut ในการตัดคำภาษาไทย

คืนค่าเป็น "list" เช่น ['แมว','กิน']

ตัวอย่าง

```
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']
```

Postaggers ภาษาไทย

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

list คือ list ที่เก็บข้อความหลังผ่านการตัดคำแล้ว

engine คือ ชุดเครื่องมือในการ postaggers มี 2 ตัวดังนี้

1. old เป็น UnigramTagger (ค่าเริ่มต้น)
2. artagger เป็น RDR POS Tagger ละเอียดยิ่งกว่าเดิม รองรับเฉพาะ Python 3 เท่านั้น

แปลงข้อความเป็น Latin

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

มี 2 engine ดังนี้

- pyicu ส่งค่า Latin
- royin ใช้หลักเกณฑ์การถอดอักษรไทยเป็นอักษรโรมัน ฉบับราชบัณฑิตยสถาน (**หากมีข้อผิดพลาด ให้ใช้คำอ่าน เนื่องจากตัว royin ไม่มีตัวแปลงคำเป็นคำอ่าน**)

data :

รับค่า "str" ข้อความ

คืนค่าเป็น "str" ข้อความ

ตัวอย่าง

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

เช็คคำผิด

ก่อนใช้งานความสามารถนี้ ให้ทำการติดตั้ง hunspell และ hunspell-th ก่อน

วิธีติดตั้ง สำหรับบน Debian , Ubuntu

```
sudo apt-get install hunspell hunspell-th
```

บน Mac OS ติดตั้งตามนี้ http://pankdm.github.io/hunspell.html

ให้ใช้ pythainlp.spell ตามตัวอย่างนี้

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

pythainlp.number

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

จัดการกับตัวเลข โดยมีดังนี้

- nttn(str) - เป็นการแปลงเลขไทยสู่เลข
- nttt(str) - เลขไทยสู่ข้อความ
- ntnt(str) - เลขสู่เลขไทย
- ntt(str) - เลขสู่ข้อความ
- ttn(str) - ข้อความสู่เลข
- numtowords(float) - อ่านจำนวนตัวเลขภาษาไทย (บาท) รับค่าเป็น "float" คืนค่าเป็น 'str'

เรียงลำดับข้อมูลภาษาไทยใน List

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

รับ list คืนค่า list

รับเวลาปัจจุบันเป็นภาษาไทย

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

WordNet ภาษาไทย

เรียกใช้งาน

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

การใช้งาน

API เหมือนกับ NLTK โดยรองรับ API ดังนี้

- 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)

ตัวอย่าง

```
>>> 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']
```

หาคำที่มีจำนวนการใช้งานมากที่สุด

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

คืนค่าออกมาเป็น dict

ตัวอย่างการใช้งาน

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

แก้ไขปัญหาการพิมพ์ลืมเปลี่ยนภาษา

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

มีคำสั่งดังนี้

- texttothai(str) แปลงแป้นตัวอักษรภาษาอังกฤษเป็นภาษาไทย
- texttoeng(str) แปลงแป้นตัวอักษรภาษาไทยเป็นภาษาอังกฤษ

คืนค่าออกมาเป็น str

Thai Character Clusters (TCC)

PyThaiNLP 1.4 รองรับ Thai Character Clusters (TCC) โดยจะแบ่งกลุ่มด้วย /

เดติด

TCC : Mr.Jakkrit TeCho

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

โค้ด : คุณ Korakot Chaovavanich

การใช้งาน

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

Enhanced Thai Char acter Cluster (ET CC)

นอกจาก TCC แล้ว PyThaiNLP 1.4 ยังรองรับ Enhanced Thai Character Cluster (ETCC) โดยแบ่งกลุ่มด้วย /

การใช้งาน

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

Thai Soundex ภาษาไทย

เดติด คุณ Korakot Chaovavanich (จาก https://gist.github.com/korakot/0b772e09340cac2f493868da035597e8)

กฎที่รองรับในเวชั่น 1.4

- กฎการเข้ารหัสซาวน์เด็กซ์ของ วิชิตหล่อจีระชุณห์กุล และ เจริญ คุวินทร์พันธุ์ - LK82
- กฎการเข้ารหัสซาวน์เด็กซ์ของ วรรณี อุดมพาณิชย์ - Udom83

การใช้งาน

```
>>> 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
```

Meta Sound ภาษาไทย

```
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
```

การใช้งาน

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

Sentiment analysis ภาษาไทย

ใช้ข้อมูลจาก https://github.com/wannaphongcom/lexicon-thai/tree/master/ข้อความ/

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

รับค่า str ส่งออกเป็น pos , neg หรือ neutral

Util

การใช้งาน

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

ngrams

สำหรับสร้าง ngrams

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

- token คือ list
- num คือ จำนวน ngrams

Corpus

stopword ภาษาไทย

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

ชื่อประเทศ ภาษาไทย

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

ตัววรรณยุกต์ในภาษาไทย

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

ตัวพยัญชนะในภาษาไทย

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

รายการคำในภาษาไทย

```
from pythainlp.corpus.thaiword import get_data # ข้อมูลเก่า
get_data()
from pythainlp.corpus.newthaiword import get_data # ข้อมูลใหม่
get_data()
```

เขียนโดย นาย วรรณพงษ์ ภัททิยไพบูลย์