Okt (Open Korean Text)

- GitHub --> https://github.com/open-korean-text/open-korean-text

-                        " "
    -        (   ) :   ,
    -        :
    -    )   .
        -         : ;   ,
        -         : ,

Okt (Open Korean Text)

-                          (twitter-korean-text)
-                          ,   ,
-       ,   ,   ,
- Scala   , Java, Python, Ruby   wrapper

Time analysis [1]

1. 로딩 시간: 사전 로딩을 포함하여 클래스를 로딩하는 시간.

    - Kkma: 5.6988 *secs*
    - Komoran: 5.4866 *secs*
    - Hannanum: 0.6591 *secs*
    - Okt (previous `Twitter`): 1.4870 *secs*
    - Mecab: 0.0007 *secs*

2. 실행시간: 10만 문자의 문서를 대상으로 각 클래스의 `pos` 메소드를 실행하는데 소요되는 시간.

    - Kkma: 35.7163 *secs*
    - Komoran: 25.6008 *secs*
    - Hannanum: 8.8251 *secs*
    - Okt (previous `Twitter`): 2.4714 *secs*
    - Mecab: 0.2838 *secs*

    문자의 개수를 늘려감에 따라 모든 클래스의 실행 시간은 기하급수적으로 증가합니다.

Okt

-     :
-     :

1.   ,   ,   ,

>> PoS [(' ', 'Noun'), ('   ', 'Noun'), ('   ', 'Noun'), ('   ', 'Noun')]

Valve >> PoS [('   ', 'Noun'), (' ', 'Modifier'), (' ', 'Modifier'), (' ', 'Noun')

>> PoS [('   ', 'Noun'), ('   ', 'Noun'), ('   ', 'Noun')

>> PoS [(' ', 'Noun'), (' ', 'Verb'), ('   ', 'Verb')

2.   ,   .

okt.morphs(
'Languagemodeling involves developing a statisticalmodel for predicting the next word in a sentence.'
>> ['Languagemodeling','involves','developing','a','statisticalmodel','for','predicting','the', 'next','word','in','a','sentence','.']

argument (stem= True)

(Lemmatization)

- Okt
-
-                              "   "   "   "
-
-   (stem) +   (affix)   parsing

) okt.pos('   ', stem= True)
>> [('   ', 'Verb')]

(Stemming)

-

)   -->   +

Okt Methods (   )

- morphs(phrase, norm= False, stem= False)

text = '   '
print(okt.morphs(text))
>> ['   ',' ',' ',' ','   ','   ','   ','   ','   ',' ',' ',' ']

print(okt.morphs(text, stem = True))
>> ['   ',' ',' ',' ',' ','   ','   ','   ','   ','   ',' ',' ',' ']

- normalize(phrase)
  ○

＊

```
text = '      .        '
okt.normalize(text)
>>          .
```

- nouns(phrase)

```
text = "                                   ."
okt.nouns(text)
>>  ['      ', '      ', '   ', '      ']
```

- phrases(phrase)

```
text = "                                              ."
okt.phrases(text)
>>  ['    ', '                 ', '                  ', '       ', '      ', '      ', '                 ', '      ', '     ']
```

- pos(phrase, norm= False, stem= False, join= False)
    - norm – If True, normalize tokens.
    - stem – If True, stem tokens.
        - ，   (didctionary)
    - join – If True, returns joined sets of morph and tag.

```
text = '                                              '
print(okt.pos(text))
>>  [('          ', 'Adjective'), ('  ', 'Noun'), ('  ', 'Josa'), ('  ', 'Noun'), ('    ', 'Verb'), ('    ',
    'Exclamation'), ('    ', 'Exclamation'), ('    ', 'Noun'), ('       ', 'Verb'), ('    ', 'Determiner'), (
    '  ', 'Noun'), ('  ', 'Josa')]
```

---

a.png   61.4 KB   2022-03-31