Kumar Gaurav 20122065

Lab 3 , 08 sept 2021

program ¶

write a program to tokenize non english write a program synonyms & antonyms from wordnet

In [1]:

```
import nltk
import os
import nltk.corpus
```

In [3]:

```
import inltk
```

In [4]:

```
from nltk.tokenize import sent_tokenize
mytext = "Bonjour M. Adam, comment allez-vous? J'espère que tout va bien. Aujourd'hui est u
print(sent_tokenize(mytext,"french"))
```

```
['Bonjour M. Adam, comment allez-vous?', "J'espère que tout va bien.", "Aujo
urd'hui est u"]
```

In [5]:

```
text = '''
NLTK ist Open Source Software. Der Quellcode wird unter den Bedingungen der Apache License
Die Dokumentation wird unter den Bedingungen der Creative Commons-Lizenz Namensnennung - Ni
abgeleiteten Werke 3.0 in den Vereinigten Staaten verteilt.
'''
print("\nOriginal string:")
print(text)
from nltk.tokenize import sent_tokenize
token_text = sent_tokenize(text, language='german')
print("\nSentence-tokenized copy in a list:")
print(token_text)
print("\nRead the list:")
for s in token_text:
    print(s)
```

```
Original string:

NLTK ist Open Source Software. Der Quellcode wird unter den Bedingungen der
Apache License Version 2.0 vertrieben.
Die Dokumentation wird unter den Bedingungen der Creative Commons-Lizenz Nam
ensnennung - Nicht kommerziell - Keine
abgeleiteten Werke 3.0 in den Vereinigten Staaten verteilt.


Sentence-tokenized copy in a list:
['\nNLTK ist Open Source Software.', 'Der Quellcode wird unter den Bedingung
en der Apache License Version 2.0 vertrieben.', 'Die Dokumentation wird unte
r den Bedingungen der Creative Commons-Lizenz Namensnennung - Nicht kommerzi
ell - Keine \nabgeleiteten Werke 3.0 in den Vereinigten Staaten verteilt.']

Read the list:

NLTK ist Open Source Software.
Der Quellcode wird unter den Bedingungen der Apache License Version 2.0 vert
rieben.
Die Dokumentation wird unter den Bedingungen der Creative Commons-Lizenz Nam
ensnennung - Nicht kommerziell - Keine
abgeleiteten Werke 3.0 in den Vereinigten Staaten verteilt.
```

In [18]:

```
from inltk.inltk import setup
setup('hi')
```

```
---------------------------------------------------------------------------
-
RuntimeError                              Traceback (most recent call las
t)
<ipython-input-18-ea8b68f1926d> in <module>
      1 from inltk.inltk import setup
----> 2 setup('hi')

~\anaconda3\lib\site-packages\inltk\inltk.py in setup(language_code)
     31     loop = asyncio.get_event_loop()
     32     tasks = [asyncio.ensure_future(download(language_code))]
---> 33     learn = loop.run_until_complete(asyncio.gather(*tasks))[0]
     34     loop.close()
     35 

~\anaconda3\lib\asyncio\base_events.py in run_until_complete(self, future)
    568         future.add_done_callback(_run_until_complete_cb)
    569         try:
--> 570             self.run_forever()
```

In [19]:

```
from inltk.inltk import tokenize
hindi_text = """प्राचीन काल में विक्रमादित्य नाम के एक आदर्श राजा हुआ करते थे।
अपने साहस, पराक्रम और शौर्य के लिए  राजा विक्रम मशहूर थे।
ऐसा भी कहा जाता है कि राजा विक्रम अपनी प्राजा के जीवन के दुख दर्द जानने के लिए रात्री के पहर में भेष बदल कर
# tokenize(input text, language code)
tokenize(hindi_text, "hi")
```

Out[19]:

```
['▁प्राचीन',
 '▁काल',
 '▁में',
 '▁विक्रमादित्य',
 '▁नाम',
 '▁के',
 '▁एक',
 '▁आदर्श',
 '▁राजा',
 '▁हुआ',
 '▁करते',
 '▁थे',
 '।',
 '▁अपने',
 '▁साहस',
 ',',
 '▁पराक्रम',
 '▁और',
 '▁शौर्य',
 '▁के',
 '▁लिए',
 '▁राजा',
 '▁विक्रम',
 '▁मशहूर',
 '▁थे',
 '।',
 '▁ऐसा',
 '▁भी',
 '▁कहा',
 '▁जाता',
 '▁है',
 '▁कि',
 '▁राजा',
 '▁विक्रम',
 '▁अपनी',
 '▁प्रा',
 'जा',
 '▁के',
 '▁जीवन',
 '▁के',
 '▁दुख',
 '▁दर्द',
 '▁जानने',
 '▁के',
 '▁लिए',
 '▁रात्री',
 '▁के',
 '▁पहर',
 '▁में',
 '▁भेष',
 '▁बदल',
 '▁कर',
 '▁नगर',
 '▁में',
 '▁घूमते',
 '▁थे',
 '।']
```

In [20]:

```
from inltk.inltk import get_similar_sentences
# get similar sentences to the one given in hindi
output = get_similar_sentences('मैं आज बहुत खुश हूं', 5, 'hi')
print(output)
```

In [21]:

```
from inltk.inltk import setup
setup('bn')
```

```
---------------------------------------------------------------------------
-
RuntimeError                              Traceback (most recent call las
t)
<ipython-input-21-a7ed8ed5b98b> in <module>
      1 from inltk.inltk import setup
----> 2 setup('bn')

~\anaconda3\lib\site-packages\inltk\inltk.py in setup(language_code)
     31     loop = asyncio.get_event_loop()
     32     tasks = [asyncio.ensure_future(download(language_code))]
---> 33     learn = loop.run_until_complete(asyncio.gather(*tasks))[0]
     34     loop.close()
     35

~\anaconda3\lib\asyncio\base_events.py in run_until_complete(self, future)
    568         future.add_done_callback(_run_until_complete_cb)
    569         try:
--> 570             self.run_forever()
```

In [22]:

```
from inltk.inltk import setup
from inltk.inltk import predict_next_words
# download models for Gujarati
setup('bn')
# predict the next words of the sentence "The weather is nice today"
predict_next_words("আবহাওয়া চমৎকার", 10, "bn", 0.7)
```

```
---------------------------------------------------------------------------
-
RuntimeError                              Traceback (most recent call las
t)
<ipython-input-22-a6ad23fec348> in <module>
      2 from inltk.inltk import predict_next_words
      3 # download models for Gujarati
----> 4 setup('bn')
      5 # predict the next words of the sentence "The weather is nice toda
y"
      6 predict_next_words("আবহাওয়া চমৎকার", 10, "bn", 0.7)

~\anaconda3\lib\site-packages\inltk\inltk.py in setup(language_code)
     31     loop = asyncio.get_event_loop()
     32     tasks = [asyncio.ensure_future(download(language_code))]
---> 33     learn = loop.run_until_complete(asyncio.gather(*tasks))[0]
     34     loop.close()
     35
```

In [23]:

```
from inltk.inltk import setup
setup('ml')
```

```
---------------------------------------------------------------------------
-
RuntimeError                              Traceback (most recent call las
t)
<ipython-input-23-1de87b760216> in <module>
      1 from inltk.inltk import setup
----> 2 setup('ml')

~\anaconda3\lib\site-packages\inltk\inltk.py in setup(language_code)
     31     loop = asyncio.get_event_loop()
     32     tasks = [asyncio.ensure_future(download(language_code))]
---> 33     learn = loop.run_until_complete(asyncio.gather(*tasks))[0]
     34     loop.close()
     35

~\anaconda3\lib\asyncio\base_events.py in run_until_complete(self, future)
    568         future.add_done_callback(_run_until_complete_cb)
    569         try:
--> 570             self.run_forever()
```

In [24]:

```
from inltk.inltk import tokenize
malayalam_text ="എതിർവർഗ്ഗലൈംഗികത (Heterosexuality), ഉഭയലൈംഗികത/ദ്വിവർഗ്ഗ
tokenize(malayalam_text, "ml")
```

Out[24]:

```
['▁എതിർ',
 'വർഗ്ഗ',
 'ലൈ',
 'ംഗ',
 'ിക',
 'ത',
 '▁',
 '(',
 'H',
 'e',
 'ter',
 'o',
 'se',
 'x',
 'u',
 'al',
 'ity',
 ')',
 ',',
 '▁ഉ',
 'ഭ',
 'യ',
 'ലൈ',
 'ംഗ',
 'ിക',
 'ത',
 '/',
 'ദ്',
 'വി',
 'വർഗ്ഗ',
 'ലൈ',
 'ംഗ',
 'ിക',
 'ത',
 '▁എന്നിവയ്ക്ക',
 'ൊപ്പം',
 '▁ലൈംഗിക',
 'തയുടെ',
 '▁തുടർച്ച',
 'യിലെ',
 '▁മൂന്നു',
 '▁തരം',
 'തിരി',
 'വു',
 'കളിലൊന്നാണ്']
```

In [28]:

```
from inltk.inltk import setup
setup('ta')
```

```
---------------------------------------------------------------------------
-
RuntimeError                              Traceback (most recent call las
t)
<ipython-input-28-2b3dd1423cc4> in <module>
      1 from inltk.inltk import setup
----> 2 setup('ta')

~\anaconda3\lib\site-packages\inltk\inltk.py in setup(language_code)
     31     loop = asyncio.get_event_loop()
     32     tasks = [asyncio.ensure_future(download(language_code))]
---> 33     learn = loop.run_until_complete(asyncio.gather(*tasks))[0]
     34     loop.close()
     35 

~\anaconda3\lib\asyncio\base_events.py in run_until_complete(self, future)
    568         future.add_done_callback(_run_until_complete_cb)
    569         try:
--> 570             self.run_forever()
```

In [29]:

```
from inltk.inltk import tokenize
tamil_text = "எனக்கு என் குழந்தைப் பருவம் நினைவிருக்கிறது"
# tokenize(input text, language code)
tokenize(tamil_text, "ta")
```

Out[29]:

```
['▁என',
 'க்கு',
 '▁என்',
 '▁குழந்தை',
 'ப்',
 '▁பருவ',
 'ம்',
 '▁',
 'நினைவ',
 '◌ிருக்கிறது']
```

In [30]:

```
from inltk.inltk import predict_next_words
# predict the next words of the sentence "The weather is nice today"
predict_next_words("আবহাওয়া চমৎকার", 10, "bn", 0.7)
```

In [31]:

```
bengali_text ="আবহাওয়া চমৎকার"
tokenize(bengali_text, "bn")
```

Out[31]:

```
['▁আবহাওয়া', '▁চমৎকার']
```

In [34]:

```
setup('pa')
```

```
---------------------------------------------------------------------------
RuntimeError                              Traceback (most recent call last)
<ipython-input-34-9685d8291d1b> in <module>
----> 1 setup('pa')

~\anaconda3\lib\site-packages\inltk\inltk.py in setup(language_code)
     31     loop = asyncio.get_event_loop()
     32     tasks = [asyncio.ensure_future(download(language_code))]
---> 33     learn = loop.run_until_complete(asyncio.gather(*tasks))[0]
     34     loop.close()
     35 

~\anaconda3\lib\asyncio\base_events.py in run_until_complete(self, future)
    568         future.add_done_callback(_run_until_complete_cb)
    569         try:
--> 570             self.run_forever()
    571         except:
    572             if new_task and future.done() and not future.cancelled()
:

~\anaconda3\lib\asyncio\base_events.py in run_forever(self)
    523         self._check_closed()
    524         if self.is_running():
--> 525             raise RuntimeError('This event loop is already running')
    526         if events._get_running_loop() is not None:
    527             raise RuntimeError(

RuntimeError: This event loop is already running


Done!
```

In [35]:

```
punjabi_txt = "ਸ਼ੁਭ ਸਵੇਰ. ਤੁਹਾਡਾ ਦਿਨ ਚੰਗਾ ਬੀਤੇ!"
tokenize(punjabi_txt, "pa")
```

Out[35]:

```
['▁ਸ਼ੁਭ', '▁ਸਵੇਰ', '.', '▁ਤੁਹਾਡਾ', '▁ਦਿਨ', '▁ਚੰਗਾ', '▁ਬੀਤੇ', '!']
```

program synonysm and antonysm

In [36]:

```
from nltk.corpus import wordnet
```

In [37]:

```
synonyms = []
for syn in wordnet.synsets("Soil"):
    for lm in syn.lemmas():
             synonyms.append(lm.name())
print (set(synonyms))
```

```
{'begrime', 'soil', 'colly', 'grunge', 'filth', 'land', 'stain', 'ground',
'territory', 'bemire', 'dirty', 'grease', 'grime', 'dirt'}
```

In [38]:

```
antonyms = []
for syn in wordnet.synsets("ahead"):
    for lm in syn.lemmas():
        if lm.antonyms():
            antonyms.append(lm.antonyms()[0].name())
print(set(antonyms))
```

```
{'backward', 'back'}
```

In [39]:

```
from inltk.inltk import get_similar_sentences
# get similar sentences to the one given in hindi
output = get_similar_sentences('मैं आज बहुत खुश हूं', 5, 'hi')
print(output)
```

In [40]:

```
pip install flair
```

```
Collecting flair
  Downloading flair-0.9-py3-none-any.whl (319 kB)
Requirement already satisfied: scikit-learn>=0.21.3 in c:\users\hp\anacond
a3\lib\site-packages (from flair) (0.24.2)
Requirement already satisfied: python-dateutil>=2.6.1 in c:\users\hp\anaco
nda3\lib\site-packages (from flair) (2.8.1)
Requirement already satisfied: tqdm>=4.26.0 in c:\users\hp\anaconda3\lib\s
ite-packages (from flair) (4.42.1)
Requirement already satisfied: lxml in c:\users\hp\anaconda3\lib\site-pack
ages (from flair) (4.5.0)
Collecting deprecated>=1.2.4
  Downloading Deprecated-1.2.13-py2.py3-none-any.whl (9.6 kB)
Collecting sentencepiece==0.1.95
  Downloading sentencepiece-0.1.95-cp37-cp37m-win_amd64.whl (1.2 MB)
Collecting gensim<=3.8.3,>=3.4.0
  Downloading gensim-3.8.3-cp37-cp37m-win_amd64.whl (24.2 MB)
Collecting ftfy
  Downloading ftfy-6.0.3.tar.gz (64 kB)
Collecting langdetect
```