jupyter Lemmatization-Technique Last Checkpoint: 40 minutes ago

File Edit View Run Kernel Settings Help Trusted

Markdown JupyterLab Python 3 (ipykernel)

Lemmatization using NLTK

[1]:
```
!pip install nltk
```

```
Requirement already satisfied: nltk in d:\iu-training\introduction-to-al-and-ml\source-code ai\spacy-package\myenv\lib\site-packages (3.9.1)
Requirement already satisfied: click in d:\iu-training\introduction-to-al-and-ml\source-code ai\spacy-package\myenv\lib\site-packages (from nltk)
(8.2.1)
Requirement already satisfied: joblib in d:\iu-training\introduction-to-al-and-ml\source-code ai\spacy-package\myenv\lib\site-packages (from nltk)
(1.5.1)
Requirement already satisfied: regex>=2021.8.3 in d:\iu-training\introduction-to-al-and-ml\source-code ai\spacy-package\myenv\lib\site-packages (fr
om nltk) (2024.11.6)
Requirement already satisfied: tqdm in d:\iu-training\introduction-to-al-and-ml\source-code ai\spacy-package\myenv\lib\site-packages (from nltk)
(4.67.1)
Requirement already satisfied: colorama in d:\iu-training\introduction-to-al-and-ml\source-code ai\spacy-package\myenv\lib\site-packages (from clic
k->nltk) (0.4.6)
```

[2]:
```
import nltk
```

[4]:
```
nltk.download('punkt')
```

```
[nltk_data] Downloading package punkt to
[nltk_data]     C:\Users\Imart\AppData\Roaming\nltk_data...
[nltk_data]   Package punkt is already up-to-date!
```

[4]:
```
True
```

[13]:
```
from nltk.stem import WordNetLemmatizer #dictionary List ---> Wordlist Donwload
```

[10]:
```
nltk.download('wordnet') # Simple dictionary
```

```
[nltk_data] Downloading package wordnet to
[nltk_data]     C:\Users\Imart\AppData\Roaming\nltk_data...
[nltk_data]   Package wordnet is already up-to-date!
```

[10]:
```
True
```

[14]:
```
nltk.download('omw-1.4') # Extended Wordlist (Huge dictionary)
```

```
[nltk_data] Downloading package omw-1.4 to
[nltk_data]     C:\Users\Imart\AppData\Roaming\nltk_data...
```

[14]:
```
True
```

[15]:
```
lemmatizer = WordNetLemmatizer()
```

[16]:
```
words = ['cats','dogs','better','running','congrats','congratulations','happily','played','ate','studies','geese']
```

[18]:
```
print('words list :',words)
```

```
words list : ['cats', 'dogs', 'better', 'running', 'congrats', 'congratulations', 'happily', 'played', 'ate', 'studies', 'geese']
```

[19]:
```
lemmatized_words = [lemmatizer.lemmatize(word.lower()) for word in words]
```

[20]:
```
print('Lemmatised words :',lemmatized_words)
```

```
Lemmatised words : ['cat', 'dog', 'better', 'running', 'congrats', 'congratulation', 'happily', 'played', 'ate', 'study', 'goose']
```

[21]:
```
print(lemmatizer.lemmatize('better')) #comparative Nouns : Best,Better,Good, : POS = 'noun'
```

```
better
```

[22]:
```
print(lemmatizer.lemmatize('cats'))
```

```
cat
```

[23]:
```
print(lemmatizer.lemmatize('running')) #it is not working because we have not post tagging. Verb POS='Verb'
```

```
running
```

How to Perform Post Tagging (Parts of Speech in Lemmatization)

[24]:
```
from nltk.corpus import wordnet
```

[25]:
```
from nltk import pos_tag,word_tokenize
```

[26]:
```
nltk.download('punkt')
```

```
[nltk_data] Downloading package punkt to
[nltk_data]     C:\Users\Imart\AppData\Roaming\nltk_data...
[nltk_data]   Package punkt is already up-to-date!
```

[26]:
```
True
```

[27]:
```
def word_tagging(word_tag):
    if word_tag.startswith('J'):
        return wordnet.ADJ
    elif word_tag.startswith('V'):
        return wordnet.VERB
    elif word_tag.startswith('N'):
        return wordnet.NOUN
    elif word_tag.startswith('R'):
        return wordnet.ADV
    else:
        return wordnet.NOUN
```

[40]:
```
corpus  = 'The Stripped Bats are hanging in thier feet of the better view'
```

[41]:
```
print('Corupus:',corpus)
```

```
Corupus: The Stripped Bats are hanging in thier feet of the better view
```

[42]:
```
tokens = word_tokenize(corpus)
```

[31]:
```
print('Word Tokens:',tokens)
```

```
Word Tokens: ['The', 'Stripped', 'Bats', 'are', 'hanging', 'in', 'thier', 'feet', 'of', 'the', 'best', 'view']
```

[34]:
```
import nltk
```

[35]:
```
nltk.download('averaged_perceptron_tagger_eng')  # Tagging Model
```

```
[nltk_data] Downloading package averaged_perceptron_tagger_eng to
[nltk_data]     C:\Users\Imart\AppData\Roaming\nltk_data...
[nltk_data]   Unzipping taggers\averaged_perceptron_tagger_eng.zip.
```

[35]:
```
True
```

[36]:
```
tagging = pos_tag(tokens)
print('Tagged words:',tagging)
```

```
Tagged words: [('The', 'DT'), ('Stripped', 'NNP'), ('Bats', 'NNP'), ('are', 'VBP'), ('hanging', 'VBG'), ('in', 'IN'), ('thier', 'JJ'), ('feet', 'NN
S'), ('of', 'IN'), ('the', 'DT'), ('best', 'JJS'), ('view', 'NN')]
```

list of the Post Tagging

[37]:
```
post_taggs = '''
POS tag list:

CC  coordinating conjunction
CD  cardinal digit
DT  determiner
EX  existential there (like: "there is" ... think of it like "there exists")
FW  foreign word
IN  preposition/subordinating conjunction
JJ  adjective   'big'
JJR adjective, comparative  'bigger'
JJS adjective, superlative  'biggest'
LS  list marker 1)
MD  modal   could, will
NN  noun, singular 'desk'
NNS noun plural 'desks'
NNP proper noun, singular   'Harrison'
NNPS    proper noun, plural 'Americans'
PDT predeterminer   'all the kids'
POS possessive ending   parent\'s
PRP personal pronoun    I, he, she
PRP$    possessive pronoun  my, his, hers
RB  adverb  very, silently,
RBR adverb, comparative better
RBS adverb, superlative best
RP  particle    give up
TO  to  go 'to' the store.
UH  interjection    errrrrrrrm
VB  verb, base form take
VBD verb, past tense    took
VBG verb, gerund/present participle taking
VBN verb, past participle   taken
VBP verb, sing. present, non-3d take
VBZ verb, 3rd person sing. present  takes
WDT wh-determiner   which
WP  wh-pronoun  who, what
WP$ possessive wh-pronoun   whose
WRB wh-abverb   where, when
'''
print(post_taggs)
```

```
POS tag list:

CC      coordinating conjunction
CD      cardinal digit
DT      determiner
EX      existential there (like: "there is" ... think of it like "there exists")
FW      foreign word
IN      preposition/subordinating conjunction
JJ      adjective       'big'
JJR     adjective, comparative  'bigger'
JJS     adjective, superlative  'biggest'
LS      list marker     1)
MD      modal   could, will
NN      noun, singular 'desk'
NNS     noun plural     'desks'
NNP     proper noun, singular   'Harrison'
NNPS    proper noun, plural     'Americans'
PDT     predeterminer   'all the kids'
POS     possessive ending       parent's
PRP     personal pronoun        I, he, she
PRP$    possessive pronoun      my, his, hers
RB      adverb  very, silently,
RBR     adverb, comparative     better
RBS     adverb, superlative     best
RP      particle        give up
TO      to      go 'to' the store.
UH      interjection    errrrrrrrm
VB      verb, base form take
VBD     verb, past tense        took
VBG     verb, gerund/present participle taking
VBN     verb, past participle   taken
VBP     verb, sing. present, non-3d     take
VBZ     verb, 3rd person sing. present  takes
WDT     wh-determiner   which
WP      wh-pronoun      who, what
WP$     possessive wh-pronoun   whose
WRB     wh-abverb       where, when
```

```
[38]: lemmatized =  [lemmatizer.lemmatize(word,word_tagging(tag)) for word,tag in tagging]
```

```
[39]: print('Lemmatized word',lemmatized)
```

```
Lemmatized word ['The', 'Stripped', 'Bats', 'be', 'hang', 'in', 'thier', 'foot', 'of', 'the', 'best', 'view']
```

```
[ ]:
```