

Practical NLTK

David Arroyo Menéndez

October 28, 2021

Practical NLTK: Summary

```
$ git clone https://github.com/davidam/python-examples.git
```

- 1 Sentiment Analysis
- 2 Detect Gender
- 3 Sentence Similarity
- 4 Text Summary
- 5 Classify Documents
- 6 Manage Words
- 7 Gramatical Trees
- 8 Extract Keywords
- 9 Disambiguation
- 10 Bots
- 11 Extending ML with Scikit

Tweets positives versus negatives

```
$ python3 tweet-sentymment.py  
$ python3 sentiment-analysis-vader.py  
$ python3 sentiment.py
```

Your name is your gender

```
$ python3 sexmachine.py  
$ python3 perceval_git_counter_sexmachine.py  
$ python3 perceval_mbox_sexmachine.py
```

Sentence Similarity

Sentence Similarity

```
$ python3 sentence-similarity.py  
$ nosetests3 test/test_sentencesimilarity.py
```

Classify Documents

Classify Documents

```
$ python3 doc-classification-ch06.py
```

Classify Newsgroups

Classify Newsgroups

```
$ python3 nltk-sklearn.py
```

Synonyms and Antonims

```
$ python3 synonyms-antonims.py
```


Singulars and Plurals

```
$ python3 stem.py  
$ nosetests3 test/test_stem.py
```

Syllables

```
$ python3 silaba.py
```

Manage Words (IV)

For some search engines, these are some of the most common, short function words, such as the, is, at, which, and on. In this case, stop words can cause problems when searching for phrases that include them

Stopwords

```
$ python3 stopwords.py  
$ nosetests3 test/test_stopwords.py
```

Lemmas: dictionary entries

```
$ python3 wordnet-lemmatizer.py  
$ nosetests3 test/test_wordnet.py:TddInPythonExample.test_syno
```

Manage Words (VI)

A bag of words is the available words in a text and keep count of how many times they appear

Bag of words

```
$ python3 bag-of-words.py
```

Unusual words

```
$ python3 code_unusual.py
```

Trees (I)

A grammatical tree is the grammatical structure about a natural language

Trees (I)

I can build a gramatic or semantic tree from a sentence

```
$ python3 semantic-tree.py
```

I can generate sentences from a gramatic

```
$ python3 howtos/generate.py
```

I can visualize a gramatic

```
$ python3 parse-tree.py
```

I can obtain bigrams, trigrams or ngrams

```
$ python3 bigrams-trigrams.py
```

```
$ nosetests3 test/test_bigrams_trigrams.py
```

Trees (II)

Trees (II)

I can print a tree from syntactic pairs

```
$ python3 code-chinker.py
```

I can build syntactic pairs from a sentence and print a tree

```
$ python3 tokenizeandtag.py
```

```
$ python3 ner.py
```

I can find sentences verb to verb with trigrams

```
$ python3 code_three_word_phrase.py
```


Corpus

```
$ python3 gutenber.py  
$ nosetests3 test/test_gutenberg.py  
$ python3 inaugural.py  
$ python3 corpus-howto-new-corpus.py
```

Keywords: rake algorithm

Keywords: rake algorithm

```
$ python3 nltk-rake.py
```

Disambiguation

```
$ python test_all_words_wsd.py
$ python test_wsd.py

# Remember synset
$ python3 wordnet-example.py
$ nosetests3 test/test_wordnet.py
```

Detect languages

Detect languages

```
$ python3 wordnet-example.py  
$ nosetests3 test/test_wordnet.py
```

Sharing data models: pickle

Sharing data models: pickle

```
$ python3 nltk-pickle.py
```

Bots

```
$ python3 eliza.py  
$ python3 chatbot.py
```

Using Scikit in NLTK (I)



Using Scikit in NLTK (II)

Using Scikit in NLTK (II)

```
$ pip3 install damenltk  
$ cd damenltk/damenltk  
$ nosetest3 test/test_svc.py  
$ nosetest3 test/test_bernoulli.py
```


Using Scikit in NLTK (III)

Discovering NLTK and Scikit in Damegender

```
$ pip3 install damegender
$ cd damegender/src/app
$ grep sklearn *
$ grep nltk *
```

Pagerank Algorithm

A webpage has an higher pagerank if has many links the links that is linking has an higher pagerank.

- Python Tutorial (Guido Van Rossum)
- El Tutorial de Python por Guido Van Rossum
- Natural Language Processing with Python
(Steven Bird, Ewan Klein, and Edward Loper)

NLTK: The Natural Language Toolkit

Edward Loper and Steven Bird

URL: <https://arxiv.org/pdf/cs/0205028> Citations: 1952

Scikit-learn: Machine Learning in Python

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel and Bertrand Thirion

URL: <http://www.jmlr.org/papers/v12/pedregosa11a.html>

Citations: 15926

The PageRank Citation Ranking: Bringing Order to the Web

Lawrence Page, Sergey Brin, Motwani Rajeev and Winograd Terry

URL: <http://ilpubs.stanford.edu:8090/422/1/1999-66.pdf>

Citations: 15691

This document is under a Creative Commons Attribution-ShareAlike 4.0 International