EBook Gratis

APRENDIZAJE nltk

Free unaffiliated eBook created from
Stack Overflow contributors.

#nltk

Tabla de contenido

Acerca de

You can share this PDF with anyone you feel could benefit from it, downloaded the latest version from: nltk

It is an unofficial and free nltk ebook created for educational purposes. All the content is extracted from Stack Overflow Documentation, which is written by many hardworking individuals at Stack Overflow. It is neither affiliated with Stack Overflow nor official nltk.

Use the content presented in this book at your own risk; it is not guaranteed to be correct nor accurate, please send your feedback and corrections to info@zzzprojects.com

Capítulo 1: Empezando con nltk

Observaciones

NLTK es una plataforma líder para la creación de programas **Python** para trabajar con datos en lenguaje humano. Proporciona interfaces fáciles de usar para más de 50 recursos corporales y léxicos como WordNet, junto con un conjunto de bibliotecas de procesamiento de texto para clasificación, tokenización, derivación, etiquetado, análisis y razonamiento semántico. y un foro de discusión activo.

El libro

El procesamiento del lenguaje natural con Python proporciona una introducción práctica a la programación para el procesamiento del lenguaje. Escrito por los creadores de NLTK, guía al lector a través de los fundamentos de escribir programas en Python, trabajar con corpus, categorizar texto, analizar estructuras lingüísticas y más. El libro se está actualizando para Python 3 y NLTK 3. (La versión original de Python 2 todavía está disponible en http://nltk.org/book_1ed).

Versiones

Historial de versiones de NLTK

Versión	Fecha de lanzamiento
3.2.4 (**más reciente**)	2017-05-21
3.2	2016-03-03
3.1	2015-10-15

Examples

Con NLTK

Puede usar NLTK (especialmente, el paquete `nltk.tokenize`) para realizar la detección de límites de oraciones:

```
import nltk
text = "This is a test. Let's try this sentence boundary detector."
text_output = nltk.tokenize.sent_tokenize(text)
print('text_output: {0}'.format(text_output))
```

Salida:

```
text_output: ['This is a test.', "Let's try this sentence boundary detector."]
```

Instalación o configuración

NLTK requiere `Python` versiones **2.7** o **3.4+.**

Estas instrucciones consideran la versión de `python` - **3.5**

- **Mac / Unix:**

  1. Instale NLTK: ejecute `sudo pip install -U nltk`
  2. Instale Numpy (opcional): ejecute `sudo pip install -U numpy`
  3. Instalación de prueba: ejecute `python` y escriba `import nltk`

  NOTA: Para versiones anteriores de Python puede ser necesario instalar setuptools (ver http://pypi.python.org/pypi/setuptools) e instalar pip (sudo easy_install pip).

- **Windows:**

  Estas instrucciones asumen que aún no tiene Python instalado en su máquina.

  Instalación binaria de 32 bits.

  1. Instale Python 3.5: http://www.python.org/downloads/ (evite las versiones de 64 bits)
  2. Instale Numpy (opcional): http://sourceforge.net/projects/numpy/files/NumPy/ (la versión que especifica pythnon3.5)
  3. Instale NLTK: http://pypi.python.org/pypi/nltk
  4. Instalación de prueba: `Start>Python35` , luego escriba `import nltk`

- **Instalación de software de terceros:**

  Por favor, consulte: https://github.com/nltk/nltk/wiki/Installing-Third-Party-Software

---

Referencia: http://www.nltk.org/install.html

Función de descarga de NLTK

Puede instalar NLTK a través de `pip` (`pip install nltk`). Después de su instalación, muchos componentes no estarán presentes y no podrá usar algunas de las funciones de NLTK.

Desde su shell de Python, ejecute la función `ntlk.download()` para seleccionar qué paquetes adicionales desea instalar utilizando la interfaz de usuario. Alternativamente, puedes usar `python -m nltk.downloader [package_name]` .

- Para descargar todos los paquetes disponibles.

```
nltk.download('all')
```

- Para descargar el paquete específico.

```
nltk.download('package-name')
```

- Para descargar todos los paquetes de carpeta específica.

```
import nltk

dwlr = nltk.downloader.Downloader()

# chunkers, corpora, grammars, help, misc,
# models, sentiment, stemmers, taggers, tokenizers
for pkg in dwlr.packages():
    if pkg.subdir== 'taggers':
        dwlr.download(pkg.id)
```

- Para descargar todos los paquetes excepto Corpora Folder.

```
import nltk

dwlr = nltk.downloader.Downloader()

for pkg in dwlr.corpora():
    dwlr._status_cache[pkg.id] = 'installed'

dwlr.download('all')
```

Instalación de NLTK con Conda.

Para instalar NLTK con `anaconda` / `conda` .

Si está utilizando Anaconda, lo más probable es que nltk ya se haya descargado en la raíz (aunque es posible que aún necesite descargar varios paquetes manualmente).

Usando `conda` :

```
conda install nltk
```

Para actualizar `nltk` usando `conda` :

```
conda update nltk
```

Con `anaconda` :

Si está utilizando varios entornos de python en anaconda, primero active el entorno en el que

desea instalar nltk. Puede comprobar el entorno activo utilizando el comando

```
conda info --envs
```

El entorno con el signo * antes de la ruta del directorio es el activo. Para cambiar el uso del ambiente activo.

```
activate <python_version>
for eg. activate python3.5
```

Ahora revise la lista de paquetes instalados en este entorno usando commnad

```
conda list
```

Si no encuentra 'nltk' en la lista, use

```
conda install -c anaconda nltk=3.2.1
```

Para más información, puede consultar https://anaconda.org/anaconda/nltk .

Para instalar mini-conda aka `conda` : http://conda.pydata.org/docs/install/quick.html

Para instalar `anaconda` : https://docs.continuum.io/anaconda/install

Términos básicos

Cuerpo

Cuerpo del texto, singular. Corpora es el plural de este. Ejemplo: una colección de revistas médicas.

Léxico

Palabras y sus significados. Ejemplo: diccionario inglés. Considere, sin embargo, que varios campos tendrán diferentes léxicos. Por ejemplo: para un inversionista financiero, el primer significado para la palabra "Bull" es alguien que tiene confianza en el mercado, en comparación con el léxico común inglés, donde el primer significado para la palabra "Bull" es un animal. Como tal, existe un léxico especial para inversores financieros, médicos, niños, mecánicos, etc.

Simbólico

Cada "entidad" que forma parte de lo que se haya dividido se basa en reglas. Por ejemplo, cada palabra es un token cuando una oración se "tokeniza" en palabras. Cada oración también puede

ser un token, si has tokenized las oraciones de un párrafo.

Lea Empezando con nltk en línea: https://riptutorial.com/es/nltk/topic/4077/empezando-con-nltk

Capítulo 2: Distribuciones de frecuencia

Introducción

Este tema se centra en el uso de la clase nltk.FreqDist ().

Examples

Distribución de frecuencia para contar las categorías léxicas más comunes

NLTK proporciona la clase FreqDist que nos permite calcular fácilmente una distribución de frecuencia dada una lista como entrada.

Aquí estamos usando una lista de parte de las etiquetas de voz (etiquetas POS) para ver qué categorías léxicas se utilizan más en el corpus marrón.

```
import nltk

brown_tagged = nltk.corpus.brown.tagged_words()
pos_tags = [pos_tag for _,pos_tag in brown_tagged]


fd = nltk.FreqDist(pos_tags)
print(fd.most_common(5))

# Out: [('NN', 152470), ('IN', 120557), ('AT', 97959), ('JJ', 64028), ('.', 60638)]
```

Podemos ver que los sustantivos son la categoría léxica más común. Se puede acceder a las Distribuciones de frecuencia como a los diccionarios. Entonces al hacer esto podemos calcular qué porcentaje de las palabras en el cuerpo marrón son sustantivos.

```
print(fd['NN'] / len(pos_tags))
# Out: 0.1313
```

Lea Distribuciones de frecuencia en línea: https://riptutorial.com/es/nltk/topic/9318/distribuciones-de-frecuencia

Capítulo 3: Etiquetado POS

Introducción

Parte del etiquetado del habla crea **tuplas** de palabras y partes del habla. Etiqueta palabras en una oración como sustantivos, adjetivos, verbos, etc. También se puede etiquetar por tiempo, y más. Estas etiquetas significan lo que significaron en tus datos de entrenamiento originales. Usted es libre de inventar sus propias etiquetas en sus datos de entrenamiento, siempre y cuando sea consistente en su uso. Los datos de entrenamiento generalmente requieren mucho trabajo para crear, por lo que normalmente se usa un corpus preexistente. Estos usualmente usan el Penn Treebank y el Brown Corpus.

Observaciones

Puntos importantes a tener en cuenta

- La **palabra** variable es una lista de fichas.
- Aunque el elemento **i** en la **palabra de** la lista es un token, el etiquetado de un solo token etiquetará cada letra de la palabra.
- nltk.tag.pos_tag_ acepta un
    - **Lista de tokens** : luego separa y etiqueta sus elementos o
    - **lista de cuerdas**
- No puede obtener la etiqueta para una palabra, en su lugar, puede ponerla dentro de una lista.
- Etiqueta POS

Examples

Ejemplo básico

```
import nltk
from nltk.tokenize import sent_tokenize, word_tokenize
text = 'We saw the yellow dog'
word = word_tokenize(text)
tag1 = nltk.pos_tag(word)
print(tag1)
```

Lea Etiquetado POS en línea: https://riptutorial.com/es/nltk/topic/10028/etiquetado-pos

Capítulo 4: Para las palabras

Introducción

Las palabras de parada son las que se utilizan principalmente como rellenos y casi no tienen un significado útil. Debemos evitar que estas palabras ocupen espacio en la base de datos o que requieran un tiempo de procesamiento valioso. Podemos hacer fácilmente una lista de palabras para ser utilizadas como palabras vacías y luego filtrar estas palabras de los datos que queremos procesar.

Examples

Filtrar las palabras de parada

NLTK tiene por defecto un grupo de palabras que considera palabras vacías. Se puede acceder a través del corpus NLTK con:

```
from nltk.corpus import stopwords
```

Para consultar la lista de palabras vacías almacenadas para el idioma inglés:

```
stop_words = set(stopwords.words("english"))
print(stop_words)
```

Ejemplo para incorporar el conjunto de palabras de parada para eliminar las palabras de parada de un texto dado:

```
from nltk.corpus import stopwords
from nltk.tokenize import word_tokenize

example_sent = "This is a sample sentence, showing off the stop words filtration."
stop_words = set(stopwords.words('english'))
word_tokens = word_tokenize(example_sent)
filtered_sentence = [w for w in word_tokens if not w in stop_words]

filtered_sentence = []

for w in word_tokens:
    if w not in stop_words:
        filtered_sentence.append(w)

print(word_tokens)
print(filtered_sentence)
```

Lea Para las palabras en línea: https://riptutorial.com/es/nltk/topic/8750/para-las-palabras

Capítulo 5: Tallo

Introducción

Detener es una especie de método de normalización. Muchas variaciones de las palabras tienen el mismo significado, excepto cuando está involucrado el tiempo. La razón por la que nos basamos es acortar la búsqueda y normalizar las oraciones. Básicamente, es encontrar la raíz de las palabras después de eliminar de él la parte verbal y tensa. Uno de los algoritmos de derivación más populares es el stemter Porter, que ha existido desde 1979.

Examples

Porter stemmer

1. Importar `PorterStemmer` e inicializar

```
from nltk.stem import PorterStemmer
from nltk.tokenize import word_tokenize
ps = PorterStemmer()
```

2. Detener una lista de palabras

```
example_words = ["python","pythoner","pythoning","pythoned","pythonly"]

for w in example_words:
    print(ps.stem(w))
```

Resultado:

```
python
python
python
python
pythonli
```

3. Detener una oración después de tokenizing.

```
 new_text = "It is important to by very pythonly while you are pythoning with python. All
pythoners have pythoned poorly at least once."

 word_tokens = word_tokenize(new_text)
 for w in word_tokens:
     print(ps.stem(w))    # Passing word tokens into stem method of Porter Stemmer
```

Resultado:

```
It
is
```

```
import
to
by
veri
pythonli
while
you
are
python
with
python
.
all
python
have
python
poorli
at
least
onc
.
```

Lea Tallo en línea: https://riptutorial.com/es/nltk/topic/8793/tallo

Capítulo 6: Tokenización

Introducción

Se refiere a la división de oraciones y palabras del cuerpo del texto en tokens de oraciones o tokens de palabras respectivamente. Es una parte esencial de la PNL, ya que muchos módulos funcionan mejor (o solo) con etiquetas. Por ejemplo, **pos_tag** necesita *etiquetas* como entrada y no palabras, para etiquetarlas por partes del habla.

Examples

Oración y tokenización de palabras del párrafo dado por el usuario

```
from nltk.tokenize import sent_tokenize, word_tokenize
example_text = input("Enter the text:  ")

print("Sentence Tokens:")
print(sent_tokenize(example_text))

print("Word Tokens:")
print(word_tokenize(example_text))
```

Lea Tokenización en línea: https://riptutorial.com/es/nltk/topic/8749/tokenizacion

Creditos

S. No	Capítulos	Contributors
1	Empezando con nltk	alvas, Ares, Community, Franck Dernoncourt, hongsy, j_4321, JGreenwell, Mike Driscoll, Preston Hager, Pythonista, RAVI, Sameer Sinha
2	Distribuciones de frecuencia	Daniel Palenicek
3	Etiquetado POS	Sameer Sinha
4	Para las palabras	Sameer Sinha
5	Tallo	hongsy, Sameer Sinha
6	Tokenización	Sameer Sinha