Linguistica Computazionale

Laboratorio in Python - IIII

Alessio Miaschi

ItaliaNLP Lab, Istituto di Linguistica Computazionale (CNR-ILC), Pisa

alessio.miaschi@ilc.cnr.it

https://alemiaschi.github.io/

http://www.italianlp.it/alessio-miaschi/

Natural Language Processing in Python

Introduzione

- Nei moduli precedenti, abbiamo visto come aprire, leggere e estrarre semplici informazioni da file di testo (e.g. espressioni regolari, divisione del testo sulla base di spazi e ritorno a capo, etc.)

- Come fare per svolgere operazioni di elaborazione del testo più avanzate?

Introduzione

- Nei moduli precedenti, abbiamo visto come aprire, leggere e estrarre semplici informazioni da file di testo (e.g. espressioni regolari, divisione del testo sulla base di spazi e ritorno a capo, etc.)

- Come fare per svolgere operazioni di elaborazione del testo più avanzate?

- Numerose librerie in Python per il NLP:
  - NLTK;
  - spaCy;
  - Gensim;
  - Stanza.

Introduzione

- Nei moduli precedenti, abbiamo visto come aprire, leggere e estrarre semplici informazioni da file di testo (e.g. espressioni regolari, divisione del testo sulla base di spazi e ritorno a capo, etc.)

- Come fare per svolgere operazioni di elaborazione del testo più avanzate?

- Numerose librerie in Python per il NLP:
  - **NLTK**;
  - spaCy;
  - Gensim;
  - Stanza.

Natural Language Toolkit (NLTK)

- **NLTK** (https://www.nltk.org/) è una delle principali librerie in Python per l'elaborazione del linguaggio naturale

- Dispone di una vasta quantità di corpora e risorse lessicali (e.g. WordNet) e metodi/funzioni per:
  - Tokenizzazione;
  - Stemming;
  - POS tagging e Parsing;
  - Classificazione;
  - etc.

Tokenizzare un testo

```
import sys
import nltk

def main(file1):
  tokens_testo = []
  sent_tokenizer = nltk.data.load("tokenizers/punkt/english.pickle")
  with open(file1, "r") as f:
    for line in f:
      frasi = sent_tokenizer.tokenize(line)
      for frase in frasi:
        tokens = nltk.word_tokenize(frase)
        tokens_testo += tokens

  print("Tokens del testo: ")
  print(tokens_testo)

main(sys.argv[1])
```

Tokenizzare un testo

Tokenizzatore

```
import sys
import nltk

def main(file1):
  tokens_testo = []
  sent_tokenizer = nltk.data.load("tokenizers/punkt/english.pickle")
  with open(file1, "r") as f:
    for line in f:
      frasi = sent_tokenizer.tokenize(line)
      for frase in frasi:
        tokens = nltk.word_tokenize(frase)
        tokens_testo += tokens

  print("Tokens del testo: ")
  print(tokens_testo)

main(sys.argv[1])
```

Tokenizzare un testo

Tokenizzatore

```
import sys
import nltk

def main(file1):
  tokens_testo = []
  sent_tokenizer = nltk.data.load("tokenizers/punkt/english.pickle")
  with open(file1, "r") as f:
    for line in f:
      frasi = sent_tokenizer.tokenize(line)
      for frase in frasi:
        tokens = nltk.word_tokenize(frase)
        tokens_testo += tokens

  print("Tokens del testo: ")
  print(tokens_testo)

main(sys.argv[1])
```

Estrazione dei bigrammi

```
import sys
import nltk
from nltk import bigrams

def main(file1):
  tokens_testo = []
  sent_tokenizer = nltk.data.load("tokenizers/punkt/english.pickle")
  with open(file1, "r") as f:
    for line in f:
      frasi = sent_tokenizer.tokenize(line)
      for frase in frasi:
        tokens = nltk.word_tokenize(frase)
        bigrammi = list(bigrams(tokens))
        print(bigrammi)

main(sys.argv[1])
```

Estrazione dei bigrammi

```
import sys
import nltk
from nltk import bigrams

def main(file1):
  tokens_testo = []
  sent_tokenizer = nltk.data.load("tokenizers/punkt/english.pickle")
  with open(file1, "r") as f:
    for line in f:
      frasi = sent_tokenizer.tokenize(line)
      for frase in frasi:
        tokens = nltk.word_tokenize(frase)
        bigrammi = list(bigrams(tokens))
        print(bigrammi)

main(sys.argv[1])
```

POS Tagging

```
import sys
import stanza

def main(file1):
  sent_tokenizer = nltk.data.load('tokenizers/punkt/italian.pickle')
  with open(file1, 'r') as  f:
    for line in f:
      frasi = sent_tokenizer.tokenize(line)
      for frase in frasi:
        tokens = nltk.word_tokenize(frase)
        tokensPOS = nltk.pos_tag(tokens)
        print(tokensPOS)

main(sys.argv[1])

# >> [('Ciao', 'NNP'), (',', ','), ('sono', 'NN'), ('Alessio', 'NNP'), ('.', '.')]
# >> [('Questa', 'NNP'), ('è', 'NNP'), ('la', 'NNP'), ('prima', 'FW'), ('lezione', 'NN'), ('del', 'NN'), ('corso',
'NN'), ('.', '.')]
```

POS Tagging

```
import sys
import stanza

def main(file1):
  sent_tokenizer = nltk.data.load('tokenizers/punkt/italian.pickle')
  with open(file1, 'r') as  f:
    for line in f:
      frasi = sent_tokenizer.tokenize(line)
      for frase in frasi:
        tokens = nltk.word_tokenize(frase)
        tokensPOS = nltk.pos_tag(tokens)
        print(tokensPOS)

main(sys.argv[1])

# >> [('Ciao', 'NNP'), (',', ','), ('sono', 'NN'), ('Alessio', 'NNP'), ('.', '.')]
# >> [('Questa', 'NNP'), ('è', 'NNP'), ('la', 'NNP'), ('prima', 'FW'), ('lezione', 'NN'), ('del', 'NN'), ('corso',
'NN'), ('.', '.')]
```

POS Tagging

```
import sys
import stanza

def main(file1):
  sent_tokenizer = nltk.data.load('tokenizers/punkt/italian.pickle')
  with open(file1, 'r') as  f:
    for line in f:
      frasi = sent_tokenizer.tokenize(line)
      for frase in frasi:
        tokens = nltk.word_tokenize(frase)
        tokensPOS = nltk.pos_tag(tokens)
        print(tokensPOS)

main(sys.argv[1])

# >> [('Ciao', 'NNP'), (',', ','), ('sono', 'NN'), ('Alessio', 'NNP'), ('.', '.')]
# >> [('Questa', 'NNP'), ('è', 'NNP'), ('la', 'NNP'), ('prima', 'FW'), ('lezione', 'NN'), ('del', 'NN'), ('corso',
'NN'), ('.', '.')]
```

Approfondire

Riferimenti:

- Learning Python (Mark Lutz, 5th Edition, O'Reilly Media)
- **Natural Language Processing with Python (Analyzing Text with the Natural Language Toolkit) (Bird, S. et al, O'Reilly Media)**
- Python ABC (Guida online)

Approfondire

Riferimenti:

- Learning Python (Mark Lutz, 5th Edition, O'Reilly Media)
- **Natural Language Processing with Python (Analyzing Text with the Natural Language Toolkit) (Bird, S. et al, O'Reilly Media)**
- Python ABC (Guida online)

- **Stanza** (https://stanfordnlp.github.io/stanza/) è una libreria in Python sviluppata dallo *Stanford NLP Group* (https://nlp.stanford.edu/)

- Dispone di una serie di tool, da poter utilizzare in una *pipeline*, al fine di poter annotare linguisticamente un testo:
  - Supporto di più di 70 lingue;
  - Particolarmente accurato nell'annotazione linguistica

Esercizi

- Scrivere un programma in python che, a partire da un testo e usando la libreria *nltk*, estragga il numero totale di frasi presenti al suo interno e calcoli la lunghezza media delle parole (in termini di caratteri)

- Scrivere un programma in python che, a partire da un testo e usando la libreria *nltk*, restituisca in output la *Type/Token Ratio* (TTR)

- Scrivere un programma in python che, a partire da un testo e usando la libreria *nltk*, estragga il nome e il verbo (lunghi almeno 4 caratteri) con frequenza massima