Python for Text Analysis 2018-2019

Lecture 12: Revising some difficult concepts
06-12-2018

Goals for today

- **Before the break:**
  - A bit more on text analysis and NLP pipelines
  - Getting to know your data: NAF XML (Assignment 4b)
  - Playing around with the PotterAPI (JSON)
    - For-loops
    - Functions
    - Error messages / debugging your code
- **After the break:** your requests
  - Continuing with the PotterAPI
  - Working on Assignment 4

A bit more about text analysis: tokenization

- **Tokenization** means splitting texts into words
- **Why** should we worry about tokenization at all?

```
keyword = "love"
message = "I love Slovenia!"
tokenized_message = ["I", "love", "Slovenia", "!"]

print(keyword in message)
print(keyword in tokenized_message)
```

A bit more about text analysis: tokenization

- **Tokenization** means splitting texts into words
- **Why** should we worry about tokenization at all?

```
keyword = "love"
message = "Slovenia is great!"
tokenized_message = ["Slovenia", "is", "great", "!"]

print(keyword in message)
print(keyword in tokenized_message)
```

A bit more about text analysis: tokenization

- **Tokenization** means splitting texts into words
- **Why** should we worry about tokenization at all?
- **Why** should we worry about tokenization using, for instance, nltk?

```
import nltk
keyword = "love"
message = "Slovenia is the country I love!"
tokenized_message1 = message.split()
tokenized_message2 = nltk.word_tokenize(message)

print(keyword in tokenized_message1)
print(keyword in tokenized_message2)
```

A bit more about text analysis: lemmatization

- **Lemmatization** means converting words to lemmas
  - **Lemmas** are the forms of words you would find in a **dictionary**
  - For instance, **love** is the lemma of *loving, loved, loves*
- **Why** should we worry about lemmatization?

```
keyword = "love"
message = "John really loves Slovenia."
tokens = ["John", "really", "loves", "Slovenia", "."]
lemmas = ["John", "really", "love", "Slovenia", "."]

print(keyword in tokens)
print(keyword in lemmas)
```

A bit more about text analysis: POS tagging

- **POS-tagging** means tagging words in context with their **part-of-speech**
  - **Common POS**: verb (*walk*), noun (*dog*), adjective (*good*)
- **Why** should we worry about POS-tagging?

```
keyword = ("love", "verb")
message1 = "I really love Slovenia."
message2 = "My love for Slovenia is great."

tokens_with_pos1 = [("I", "pronoun",),
                    ("really", "adverb"),
                    ("love", "verb"),
                    ("Slovenia", "noun")]
tokens_with_pos2 = [("My", "pronoun",),
                    ("love", "noun"),
                    ("for", "preposition"),
                    ("Slovenia", "noun"),
                    ("is", "verb"),
                    ("great", "adjective")]

print(keyword in tokens_with_pos1)
print(keyword in tokens_with_pos2)
```

More advanced text analysis

- **Assignment 4b:** the **NAF XML file** represents the output of a complete NLP pipeline (natural language processing), which includes:
  - Tokenization
  - POS-tagging
  - Lemmatization
  - Word Sense Disambiguation → detecting the meaning of words in context (e.g. *bank*)
  - Entity Detection → recognizing Named Entities in text (persons, organizations, locations)
  - Entity Linking → linking these entities to the corresponding Wikipedia pages
    - Ford can refer to:
      https://en.wikipedia.org/wiki/Harrison_Ford
      https://en.wikipedia.org/wiki/Ford_Motor_Company

Let’s look at some data & code

- Getting to know your data: NAF XML (Assignment 4b)
- Playing around with the PotterAPI (JSON)
  - For-loops
  - Functions
  - Error messages / debugging your code

This week

- Deadline Assignment 4: **Friday 7 December at 23:59**
- **Reminder about sending your code snippets:**
  - When sharing your code snippets, please use our e-mail addresses:
    - cm.vanson@gmail.com / c.m.van.son@vu.nl
    - pia.sommerauer@vu.nl
  - E-mailing both of us has the best chance of getting a quick reply
  - Please don't share your code in a screenshot, but copy it in the e-mail or attach the actual code (notebook, .py file or .txt file)