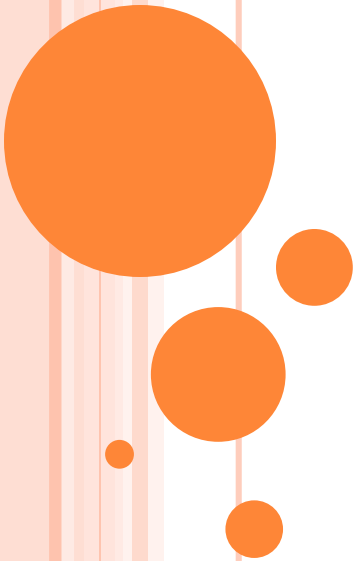


Python para PLN

Introdução ao NLTK

Roney Lira de Sales Santos roneysantos@usp.br

Prof. Thiago A. S. Pardo

NLTK

- Biblioteca de ferramentas úteis para a utilização dos princípios de PLN
- Linguagem Python
- Funcionalidades para manipulação de strings
- Interfaces padrões para realizar tarefas como etiquetar textos, frequência de palavras, lematização e stemização de palavras, entre vários outros.
- LIVRO ONLINE GRATUITO! http://www.nltk.org/book/

NLTK - Instalação

- Requer pelo menos a versão 3.5 do Python

- Linux, MacOS e Windows 32-bit
  - Para instalação no Windows 64-bit, seguir esse tutorial.

- Deu certo? Testar com o comando **`import nltk`**.

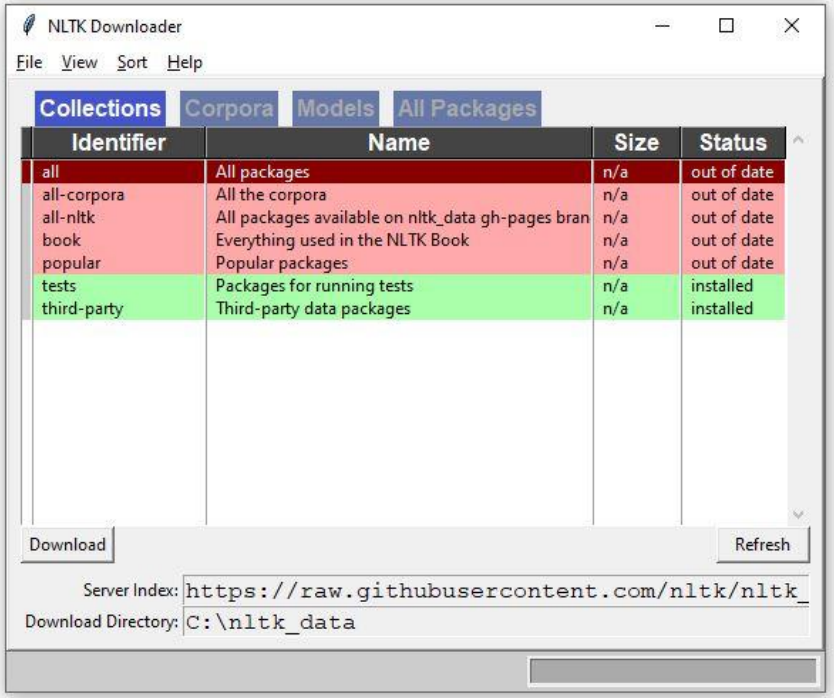
```
>>> import nltk
>>>
```

- Após a instalação do *toolkit*, instalar os dados necessários para o funcionamento.
  - NLTK data

NLTK - Instalação

- Verificar quais pacotes estão desatualizados e clicar em **Download**.
  - Selecionar o identificador **all** e clicar em **Download**.

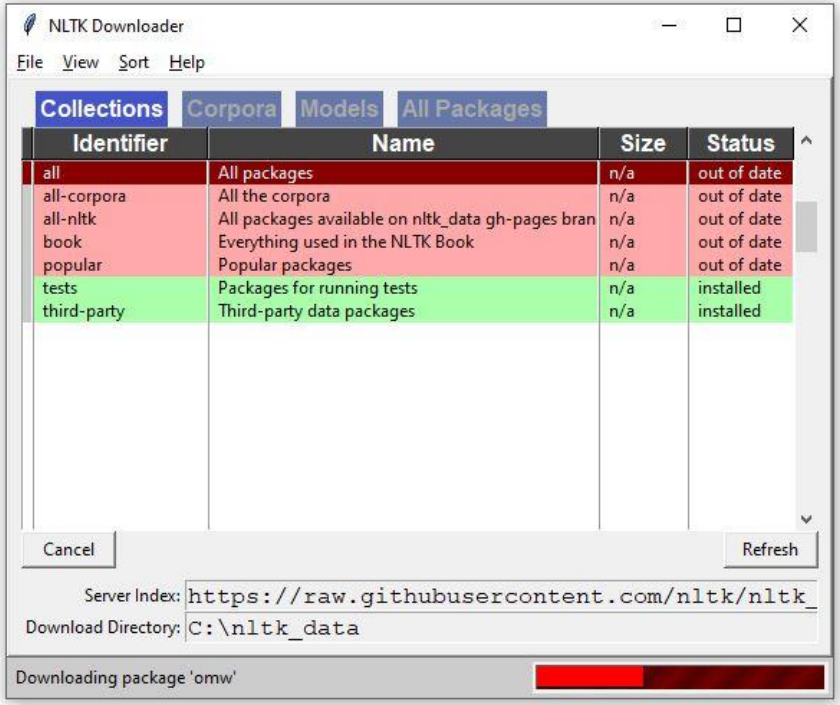
```
>>> import nltk
>>> nltk.download()
showing info https://raw.githubusercontent.com/nltk/nltk_data/gh-pages/index.xml
```



NLTK - Instalação

- A barra de progresso abaixo da janela mostra o andamento do processo.

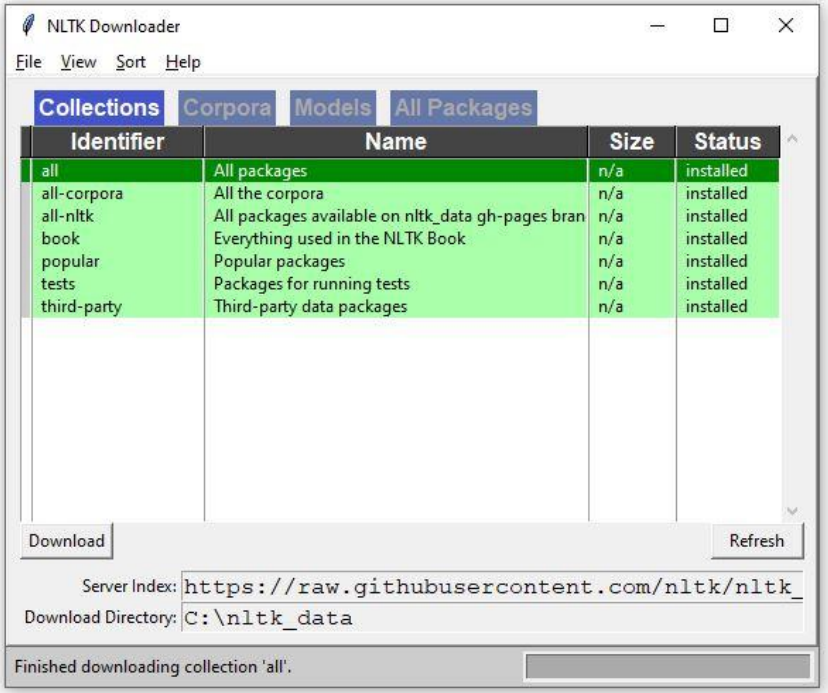
```
>>> import nltk
>>> nltk.download()
showing info https://raw.githubusercontent.com/nltk/nltk_data/gh-pages/index.xml
```



NLTK - Instalação

- A atualização demora por volta de **1 minuto e meio**. Quando estiver tudo certo, todos os pacotes estarão com a cor verde.

```
>>> import nltk
>>> nltk.download()
showing info https://raw.githubusercontent.com/nltk/nltk_data/gh-pages/index.xml
```



NLTK - Uso

- Dentro do NLTK contém **vários corpora**
  - Úteis para os etiquetadores, entidades nomeadas, estruturas sintáticas e várias outras funcionalidades.

Corpus	Compiler	Contents
Brown Corpus	Francis, Kucera	15 genres, 1.15M words, tagged, categorized
CESS Treebanks	CLiC-UB	1M words, tagged and parsed (Catalan, Spanish)
Chat-80 Data Files	Pereira & Warren	World Geographic Database
CMU Pronouncing Dictionary	CMU	127k entries
CoNLL 2000 Chunking Data	CoNLL	270k words, tagged and chunked
CoNLL 2002 Named Entity	CoNLL	700k words, pos- and named-entity-tagged (Dutch, Spanish)
CoNLL 2007 Dependency Treebanks (sel)	CoNLL	150k words, dependency parsed (Basque, Catalan)
Dependency Treebank	Narad	Dependency parsed version of Penn Treebank sample
FrameNet	Fillmore, Baker et al	10k word senses, 170k manually annotated sentences
Floresta Treebank	Diana Santos et al	9k sentences, tagged and parsed (Portuguese)
Gazetteer Lists	Various	Lists of cities and countries
Genesis Corpus	Misc web sources	6 texts, 200k words, 6 languages
Gutenberg (selections)	Hart, Newby, et al	18 texts, 2M words
Inaugural Address Corpus	CSpan	US Presidential Inaugural Addresses (1789-present)
Indian POS-Tagged Corpus	Kumaran et al	60k words, tagged (Bangla, Hindi, Marathi, Telugu)
MacMorpho Corpus	NILC, USP, Brazil	1M words, tagged (Brazilian Portuguese)
Movie Reviews	Pang, Lee	2k movie reviews with sentiment polarity classification
Names Corpus	Kantrowitz, Ross	8k male and female names
NIST 1999 Info Extr (selections)	Garofolo	63k words, newswire and named-entity SGML markup
Nombank	Meyers	115k propositions, 1400 noun frames
NPS Chat Corpus	Forsyth, Martell	10k IM chat posts, POS-tagged and dialogue-act tagged

Disponível em https://www.nltk.org/book/ch02.html

NLTK - Uso

- Dentro do NLTK contém **vários corpora**
  - Úteis para os etiquetadores, entidades nomeadas, estruturas sintáticas e várias outras funcionalidades.

```
>>> nltk.corpus.mac_morpho.words()
['Jersei', 'atinge', 'média', 'de', 'Cr$', '1,4', ...]
>>> nltk.corpus.mac_morpho.sents()
[['Jersei', 'atinge', 'média', 'de', 'Cr$', '1,4', 'milhão', 'em', 'a', 'venda', '
de', 'a', 'Pinhal', 'em', 'São', 'Paulo'], ['Programe', 'sua', 'viagem', 'a', 'a',
'Exposição', 'Nacional', 'do', 'Zebu', ',', 'que', 'começa', 'dia', '25'], ...]
>>> nltk.corpus.mac_morpho.tagged_words()
[('Jersei', 'N'), ('atinge', 'V'), ('média', 'N'), ...]
>>> nltk.corpus.mac_morpho.tagged_sents()
[[('Jersei', 'N'), ('atinge', 'V'), ('média', 'N'), ('de', 'PREP'), ('Cr$', 'CUR')
, ('1,4', 'NUM'), ('milhão', 'N'), ('em', 'PREP|+'), ('a', 'ART'), ('venda', 'N'),
('de', 'PREP|+'), ('a', 'ART'), ('Pinhal', 'NPROP'), ('em', 'PREP'), ('São', 'NPRO
P'), ('Paulo', 'NPROP')], [('Programe', 'V'), ('sua', 'PROADJ'), ('viagem', 'N'),
('a', 'PREP|+'), ('a', 'ART'), ('Exposição', 'NPROP'), ('Nacional', 'NPROP'), ('do
', 'NPROP'), ('Zebu', 'NPROP'), (',', ','), ('que', 'PRO-KS-REL'), ('começa', 'V')
, ('dia', 'N'), ('25', 'N|AP')], ...]
```

- Veremos como fazer isso para um texto qualquer!

NLTK - Uso

- Nessa aula, vamos ver detalhadamente algumas funções importantes no tratamento de textos com NLTK.

  - Tokenização
  - Frequência/Contagem de palavras
  - *Stopwords*
  - N-gramas
  - Stemmer e Lemma
  - Etiquetadores

- Para testar as funções, utilizaremos esse corpus!

NLTK - Tokenização

- **Tokenizar = separar** as palavras do texto
  - Tipo um **`split`**?

- Nível linguístico lexical: uma palavra, número ou pontuação agora é um **token**.

- Dado o texto que vai ser tokenizado, basta usar a função **`nltk.word_tokenize(texto)`**:

```
>>> import nltk
>>> texto = "O jogador, que está de camisa verde, marcou o gol da vitória!"
>>> nltk.word_tokenize(texto)
['O', 'jogador', ',', 'que', 'está', 'de', 'camisa', 'verde', ',', 'marcou'
, 'o', 'gol', 'da', 'vitória', '!']
```

NLTK - Tokenização

- O tokenizador do NLTK pode ter algumas variações, como por exemplo, retornar apenas os tokens sem as pontuações:
  - Entramos em um novo mundo chamado **expressões regulares**.

```
>>> from nltk.tokenize import RegexpTokenizer
>>> tokenizer = RegexpTokenizer(r'\w+')
>>> tokens = tokenizer.tokenize(texto)
>>> tokens
['O', 'jogador', 'que', 'está', 'de', 'camisa', 'verde', 'marcou', 'o', 'go
l', 'da', 'vitória']
```

  - E se quiséssemos os tokens sem pontuações e numerais?

```
>>> texto = "O jogador, que está com a camisa 10, marcou o gol da vitória!"
>>> from nltk.tokenize import RegexpTokenizer
>>> tokenizer = RegexpTokenizer(r'[A-z]\w*')
>>> tokens = tokenizer.tokenize(texto)
>>> tokens
['O', 'jogador', 'que', 'está', 'com', 'a', 'camisa', 'marcou', 'o', 'gol',
'da', 'vitória']
```


Mais de expressões regulares aqui nesse link.

NLTK – Frequência/Contagem

- Com a lista de tokens, é possível fazer a contagem de ocorrência de tokens pelo NLTK.
- Uso da classe **`FreqDist()`**
  - A função **`most_common()`** ordena a frequência dos tokens. Pode ser usado um argumento para informar a quantidade de tokens mais comuns.

```
>>> frequencia = nltk.FreqDist(tokens)
>>> frequencia.most_common()
[(',', 6), ('o', 4), ('a', 3), ('de', 2), ('do', 2), ('New', 2), ('e', 2),
('no', 2), ('Super', 2), ('Bowl', 2), ('.', 2), ('uma', 2), ('da', 2), ('hi
stória', 2), ('Com', 1), ('um', 1), ('passe', 1), ('Eli', 1), ('Manning', 1
), ('para', 1), ('Plaxico', 1), ('Burress', 1), ('39', 1), ('segundos', 1),
('fim', 1), ('York', 1), ('Giants', 1), ('anotou', 1), ('touchdown', 1), ('
decisivo', 1), ('derrubou', 1), ('favorito', 1), ('England', 1), ('Patriots
', 1), ('por', 1), ('17', 1), ('14', 1), ('neste', 1), ('domingo', 1), ('em
', 1), ('Glendale', 1), ('XLII', 1), ('O', 1), ('resultado', 1), ('das', 1)
, ('maiores', 1), ('zebras', 1), ('acabou', 1), ('com', 1), ('temporada', 1
), ('perfeita', 1), ('Tom', 1), ('Brady', 1), ('companhia', 1), ('que', 1),
('esperavam', 1), ('fazer', 1), ('ao', 1), ('levantar', 1), ('troféu', 1),
('NFL', 1), ('sem', 1), ('sofrer', 1), ('derrota', 1), ('ano', 1)]
>>> frequencia.most_common(5)
[(',', 6), ('o', 4), ('a', 3), ('de', 2), ('do', 2)]
```

NLTK – Frequência/Contagem

- É importante notar que os tokens em maiúsculo e minúsculo são considerados diferentes.
- Portanto, caso o objetivo da contagem seja de palavras iguais, por exemplo, é necessário usar as funções **`lower()`** (ou **`upper()`**) para normalizar os tokens.
  - Um novo princípio: **list comprehesion**

```
>>> frequencia = nltk.FreqDist(w.lower() for w in tokens)
>>> frequencia.most_common()
[(',', 6), ('o', 5), ('a', 3), ('com', 2), ('de', 2), ('do', 2), ('new', 2)
, ('e', 2), ('no', 2), ('super', 2), ('bowl', 2), ('.', 2), ('uma', 2), ('d
a', 2), ('história', 2), ('um', 1), ('passe', 1), ('eli', 1), ('manning', 1
), ('para', 1), ('plaxico', 1), ('burress', 1), ('39', 1), ('segundos', 1),
('fim', 1), ('york', 1), ('giants', 1), ('anotou', 1), ('touchdown', 1), ('
decisivo', 1), ('derrubou', 1), ('favorito', 1), ('england', 1), ('patriots
', 1), ('por', 1), ('17', 1), ('14', 1), ('neste', 1), ('domingo', 1), ('em
', 1), ('glendale', 1), ('xlii', 1), ('resultado', 1), ('das', 1), ('maiore
s', 1), ('zebras', 1), ('acabou', 1), ('temporada', 1), ('perfeita', 1), ('
tom', 1), ('brady', 1), ('companhia', 1), ('que', 1), ('esperavam', 1), ('f
azer', 1), ('ao', 1), ('levantar', 1), ('troféu', 1), ('nfl', 1), ('sem', 1
), ('sofrer', 1), ('derrota', 1), ('ano', 1)]
```

Mais de list comprehension aqui e aqui!