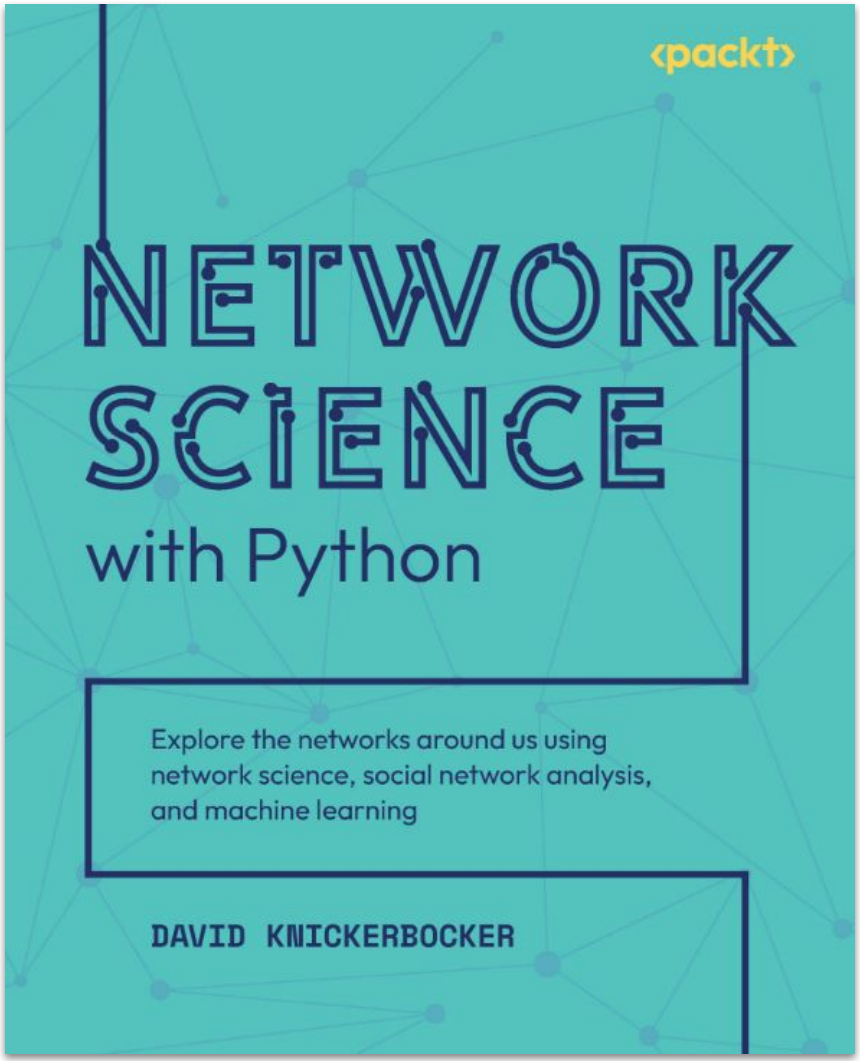
Trabalho Final



Part 2: Graph Construction and Cleanup

4

NLP and Network Synergy 89

Technical requirements 90
Why are we learning about NLP in a network book? 91
Asking questions to tell a story 91
Introducing web scraping 93
Introducing BeautifulSoup 93
Loading and scraping data with BeautifulSoup 93
Choosing between libraries, APIs, and source data 106
Using NLTK for PoS tagging 107
Using spaCy for PoS tagging and NER 121
SpaCy PoS tagging 124
SpaCy NER 127
Converting entity lists into network data 132
Converting network data into networks 136
Doing a network visualization spot check 137
Additional NLP and network considerations 147
Data cleanup 147
Comparing PoS tagging and NER 147
Scraping considerations 148
Summary 148

https://github.com/PacktPublishing/Network-Science-with-Python

Project Gutenberg

About ▾ | Search and Browse ▾ | Help ▾

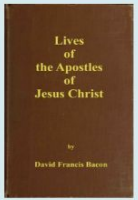
Quick search Go! | Donation PayPal

Welcome to Project Gutenberg

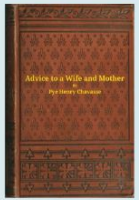
Project Gutenberg is a library of over 70,000 free eBooks

Choose among free epub and Kindle eBooks, download them or read them online. You will find the world's great literature here, with focus on older works for which U.S. copyright has expired. Thousands of volunteers digitized and diligently proofread the eBooks, for you to enjoy.

Chants for the Boer by Joaquin Miller

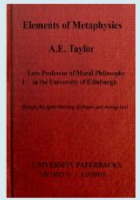


Lives of the apostles of Jesus Christ

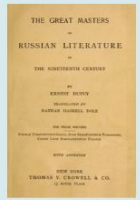


Advice to a wife and mother in two

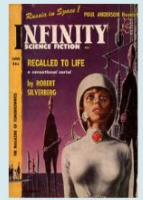
Wings of the phoenix by John Bernard



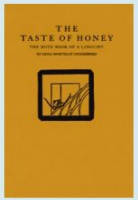
Elements of Metaphysics by A. E. Taylor



The great masters of Russian



The high ones by Poul Anderson



The taste of honey by Edna Worthley

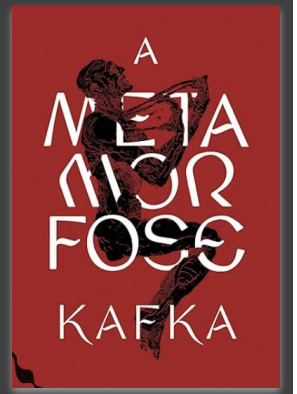
The magazine of history with notes and

Pinocchio under the sea by Gemma

Some of our latest eBooks Click Here for more latest books!

https://www.gutenberg.org

```python
import requests

# URL of the file you want to download
url = "http://www.gutenberg.org/cache/epub/5200/pg5200.txt"

# Send an HTTP GET request to the URL
response = requests.get(url)

# Check if the request was successful (status code 200)
if response.status_code == 200:
    # Open a local file to save the response content
    with open("pg5200.txt", "wb") as file:
        file.write(response.content)
    print("Download completed successfully. The file has been saved as 'pg5200.txt'")
else:
    print(f"Error downloading the file. Status code: {response.status_code}")
```



pg5200.txt

```
The Project Gutenberg eBook of Metamorphosis

This ebook is for the use of anyone anywhere in the United States and
most other parts of the world at no cost and with almost no restrictions
whatsoever. You may copy it, give it away or re-use it under the terms
of the Project Gutenberg License included with this ebook or online
at www.gutenberg.org. If you are not located in the United States,
you will have to check the laws of the country where you are located
before using this eBook.

*** This is a COPYRIGHTED Project Gutenberg eBook. Details Below. ***
***     Please follow the copyright guidelines in this file.      ***


Title: Metamorphosis


Author: Franz Kafka

Translator: David Wyllie

Release date: August 17, 2005 [eBook #5200]
                Most recently updated: April 28, 2021

Language: English



*** START OF THE PROJECT GUTENBERG EBOOK METAMORPHOSIS ***
```

```
# rename the file
!mv pg5200.txt metamorphosis.txt

# delete lines 1 to 44
!sed -i '1,44d' metamorphosis.txt

# delete lines 1861 to 2225
!sed -i '1861,2225d' metamorphosis.txt
```

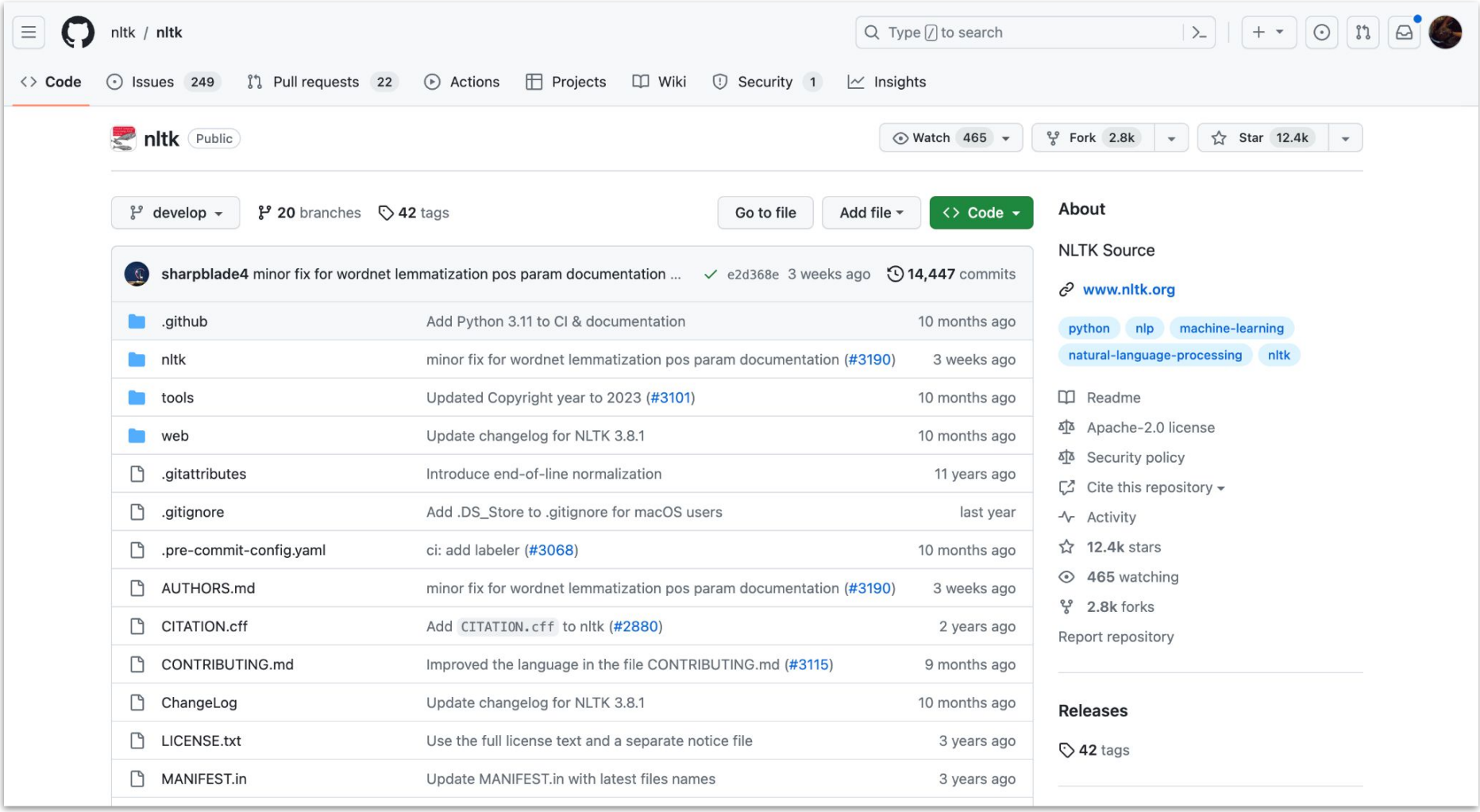
```
# load text
filename = 'metamorphosis.txt'
file = open(filename, 'rt')
text = file.read()
file.close()
```

Text Cleaning is a task-specific

- Plain text, no markup.
- Translated from German to UK English.
- Text lines break every 70 characters.
- Correct punctuation, hyphens, and names like "Mr. Samsa."

```
1 text
```

```
'One morning, when Gregor Samsa woke from troubled dreams, he found\nhimself transformed in his bed into a horrible vermin. He lay on his\narmour-like back, and if he lifted his head a little he could see his\n brown belly, slightly domed and divided by arches into stiff sections.\nThe bedding was hardly able to cover it and seemed ready to slide off\nany moment. His many legs, pitifully thin compared with the size of the\nrest of him, waved about helplessly as he looked.\n\n“What’s happened to me?” he thought. It wasn’t a dream. His room, a\nproper human room although a little too small, lay peacefully between\nits four familiar walls. A collection of textile samples lay spread out\non the table—Samsa was a travelling salesman—and above it there hung a\npicture that he had recently cut out of an illustrated magazine and\nhoused in a nice, gilded frame. It showed a lady fitted out with a fur\nhat and fur boa who sat upright, raising a heavy fur muff that covered\nthe whole of her lower a...'
```


nltk / nltk
Type / to search
Code
Issues 249
Pull requests 22
Actions
Projects
Wiki
Security 1
Insights
nltk
Public
Watch 465
Fork 2.8k
Star 12.4k
develop
20 branches
42 tags
Go to file
Add file
Code
sharpblade4 minor fix for wordnet lemmatization pos param documentation ...
e2d368e 3 weeks ago
14,447 commits
.github Add Python 3.11 to CI & documentation 10 months ago
nltk minor fix for wordnet lemmatization pos param documentation (#3190) 3 weeks ago
tools Updated Copyright year to 2023 (#3101) 10 months ago
web Update changelog for NLTK 3.8.1 10 months ago
.gitattributes Introduce end-of-line normalization 11 years ago
.gitignore Add .DS_Store to .gitignore for macOS users last year
.pre-commit-config.yaml ci: add labeler (#3068) 10 months ago
AUTHORS.md minor fix for wordnet lemmatization pos param documentation (#3190) 3 weeks ago
CITATION.cff Add CITATION.cff to nltk (#2880) 2 years ago
CONTRIBUTING.md Improved the language in the file CONTRIBUTING.md (#3115) 9 months ago
ChangeLog Update changelog for NLTK 3.8.1 10 months ago
LICENSE.txt Use the full license text and a separate notice file 3 years ago
MANIFEST.in Update MANIFEST.in with latest files names 3 years ago
About
NLTK Source
www.nltk.org
python
nlp
machine-learning
natural-language-processing
nltk
Readme
Apache-2.0 license
Security policy
Cite this repository
Activity
12.4k stars
465 watching
2.8k forks
Report repository
Releases
42 tags

Tokenization and Cleaning with NLTK

```
!pip install -U nltk
```

```
import nltk
nltk.download('gutenberg')
```

```
!python -m nltk.downloader all
```

```
import nltk
nltk.__version__
3.9.1
```

```
from nltk import sent_tokenize

# load data
filename = 'metamorphosis.txt'
file = open(filename, 'rt')
text = file.read()
file.close()

# split into sentences
sentences = sent_tokenize(text)
print(sentences[0])
```

```
One morning, when Gregor Samsa woke from troubled
dreams, he found himself transformed in his bed into
a horrible vermin.
```

- Using the Natural Language Toolkit (**NLTK**) library for part-of-speech (**PoS**) tagging
- Using **spaCy** for PoS tagging and named-entity recognition (**NER**)
- Converting entity lists into network data
- Converting network data into networks
- Doing a network visualization spot check

```
import nltk
nltk.download('punkt')
nltk.download('averaged_perceptron_tagger')

sentence = "John visited the Andes with his American friends."
tokens = nltk.word_tokenize(sentence)
pos_tags = nltk.pos_tag(tokens)

print(pos_tags)

[('John', 'NNP'), ('visited', 'VBD'), ('the', 'DT'), ('Andes', 'NNPS'),
('with', 'IN'), ('his', 'PRP$'), ('American', 'JJ'), ('friends', 'NNS')]
```

https://www.ling.upenn.edu/courses/Fall_2003/ling001/penn_treebank_pos.html

Number	Tag	Description
1.	CC	Coordinating conjunction
2.	CD	Cardinal number
3.	DT	Determiner
4.	EX	Existential *there*
5.	FW	Foreign word
6.	IN	Preposition or subordinating conjunction
7.	JJ	Adjective
8.	JJR	Adjective, comparative
9.	JJS	Adjective, superlative
10.	LS	List item marker
11.	MD	Modal
12.	NN	Noun, singular or mass
13.	NNS	Noun, plural
14.	NNP	Proper noun, singular
15.	NNPS	Proper noun, plural
16.	PDT	Predeterminer
17.	POS	Possessive ending
18.	PRP	Personal pronoun
19.	PRP$	Possessive pronoun
20.	RB	Adverb
21.	RBR	Adverb, comparative
22.	RBS	Adverb, superlative
23.	RP	Particle
24.	SYM	Symbol
25.	TO	*to*
26.	UH	Interjection
27.	VB	Verb, base form
28.	VBD	Verb, past tense
29.	VBG	Verb, gerund or present participle
30.	VBN	Verb, past participle

Substantivos (Nouns):

- **NN**: Substantivo comum no singular ou não contável.
  Exemplo: "dog", "water", "car".
- **NNS**: Substantivo comum no plural.
  Exemplo: "dogs", "cars", "houses".
- **NNP**: Substantivo próprio no singular.
  Exemplo: "John", "Brazil", "Google".
- **NNPS**: Substantivo próprio no plural.
  Exemplo: "Americans", "Andes", "Oscars".

Verbos (Verbs):

- **VB**: Verbo base (forma infinitiva).
  Exemplo: "run", "eat", "be".
- **VBD**: Verbo no passado.
  Exemplo: "ran", "ate", "was".
- **VBG**: Gerúndio ou particípio presente.
  Exemplo: "running", "eating".
- **VBN**: Particípio passado.
  Exemplo: "run", "eaten", "been".
- **VBP**: Verbo no presente, não na terceira pessoa singular.
  Exemplo: "run", "eat".
- **VBZ**: Verbo no presente, terceira pessoa singular.
  Exemplo: "runs", "eats", "is".

Conjunções (Conjunctions):

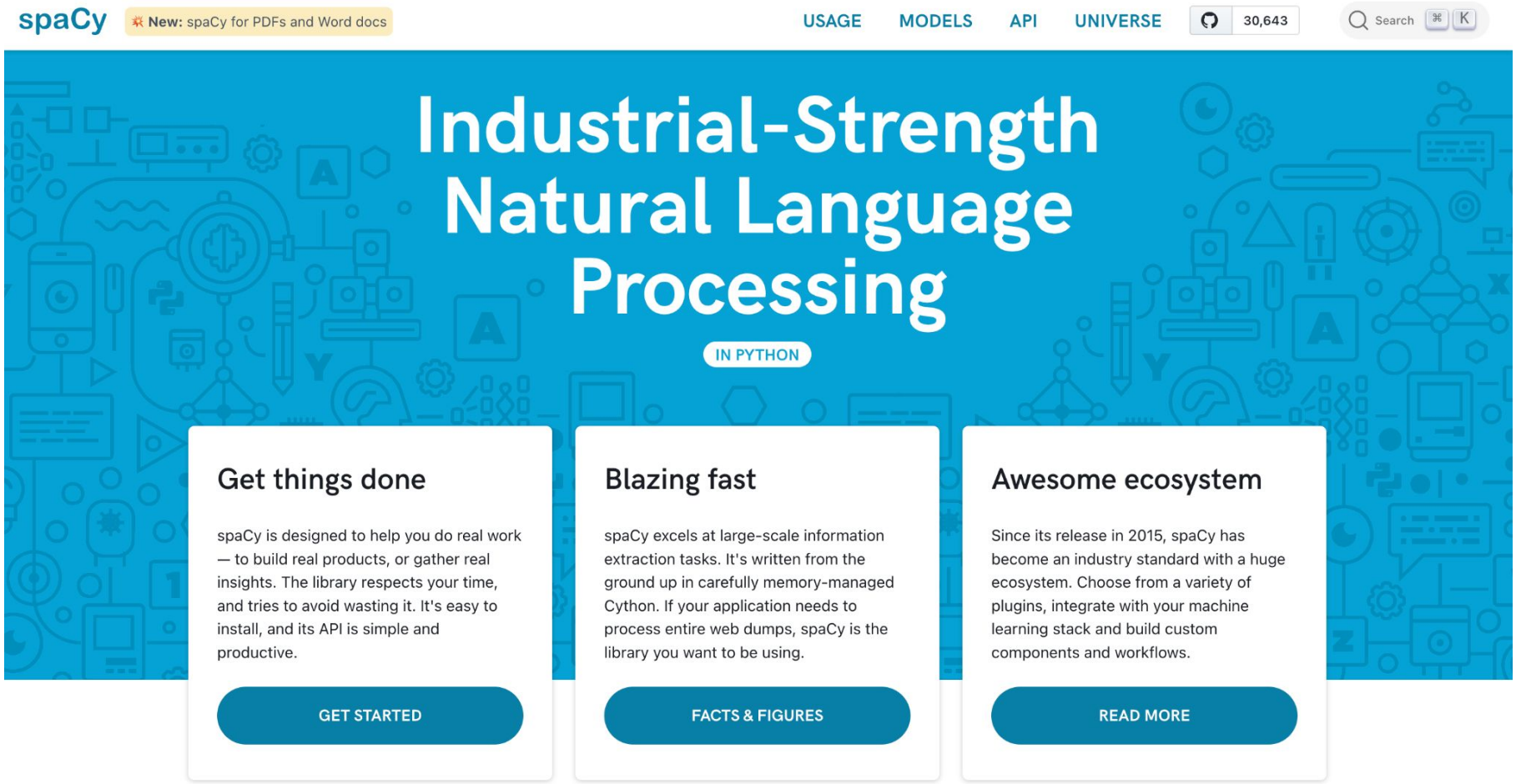
- **CC**: Conjunção coordenativa.
  Exemplo: "and", "but", "or".
- **IN**: Preposição ou conjunção subordinativa.
  Exemplo: "in", "on", "that", "because".

Adjetivos (Adjectives):

- **JJ**: Adjetivo básico.
  Exemplo: "big", "beautiful", "happy".
- **JJR**: Adjetivo no grau comparativo.
  Exemplo: "bigger", "more beautiful".
- **JJS**: Adjetivo no grau superlativo.
  Exemplo: "biggest", "most beautiful".

	sentence	entities
0	One morning, when Gregor Samsa woke from troub...	[Gregor, Samsa]
14	“Oh, God”, he thought, “what a strenuous caree...	[Oh, God]
31	“God in Heaven!” he thought.	[God, Heaven]
51	Gregor had wanted to give a full answer and e...	[Gregor, Gregor]
53	“Gregor, Gregor”, he called, “what’s wrong?” ...	[Gregor, Gregor, Gregor]
...	...	...
711	“What is it you want then?”, asked Mrs. Samsa,...	[Mrs, Samsa]
713	That’s all been sorted out.” Mrs. Samsa and Gr...	[Mrs, Samsa, Grete, Mr, Samsa]
715	“Tonight she gets sacked”, said Mr. Samsa, but...	[Tonight, Mr, Samsa]
717	Mr. Samsa twisted round in his chair to look a...	[Mr, Samsa]
727	With all the worry they had been having of lat...	[Mr, Mrs, Samsa]

74 rows × 2 columns



https://spacy.io/

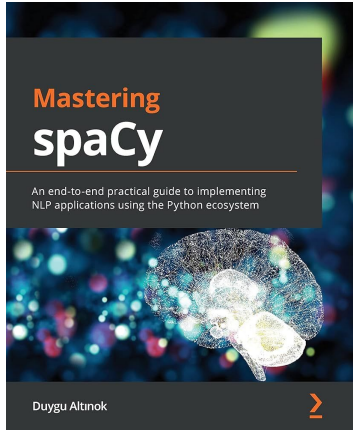
```
!python -m spacy download en_core_web_md
!python -m spacy download pt_core_news_sm


import spacy
nlp = spacy.load("en_core_web_md")
```

```python
import spacy

# Load the English language model
nlp = spacy.load("en_core_web_sm")

# Input text
text = """
One morning, when Gregor Samsa woke from troubled dreams, he found
himself transformed in his bed into a horrible vermin. He lay on his
armour-like back, and if he lifted his head a little he could see his
brown belly, slightly domed and divided by arches into stiff sections.
"""

# Process the text
doc = nlp(text)

# Extract sentences
sentences = list(doc.sents)

# Tokenize and display tags for the first sentence
for token in sentences[0]:
    print(f"{token.text}: {token.tag_}")
```

```
One: CD
morning: NN
,: ,
when: WRB
Gregor: NNP
Samsa: NNP
woke: VBD
from: IN
troubled: JJ
dreams: NNS
,: ,
he: PRP
found: VBD
himself: PRP
transformed: VBD
in: IN
his: PRP$
bed: NN
into: IN
a: DT
horrible: JJ
vermin: NN
.: .
```

PoS Tagging vs NER

The difference between PoS tagging and NER is that NER goes a step further and identifies people, places, things, and more.



Lets limit our entities to PERSON, ORG, and GPE.

```
morph_entities = extract_entities(text)
morph_entities
...
[['Gregor', 'Grete'],
['Gregor', 'Grete'],
['Grete', 'Gregor'],
['Gregor', 'Grete'],
['Grete', 'Gregor'],
['Grete', 'Gregor'],
['Grete', 'Gregor'],
['Samsa', 'Gregor'],
['Samsa', 'Gregor'],
['Samsa', 'Grete'],
['Samsa', 'Grete'],
['Samsa', 'Grete']]
```

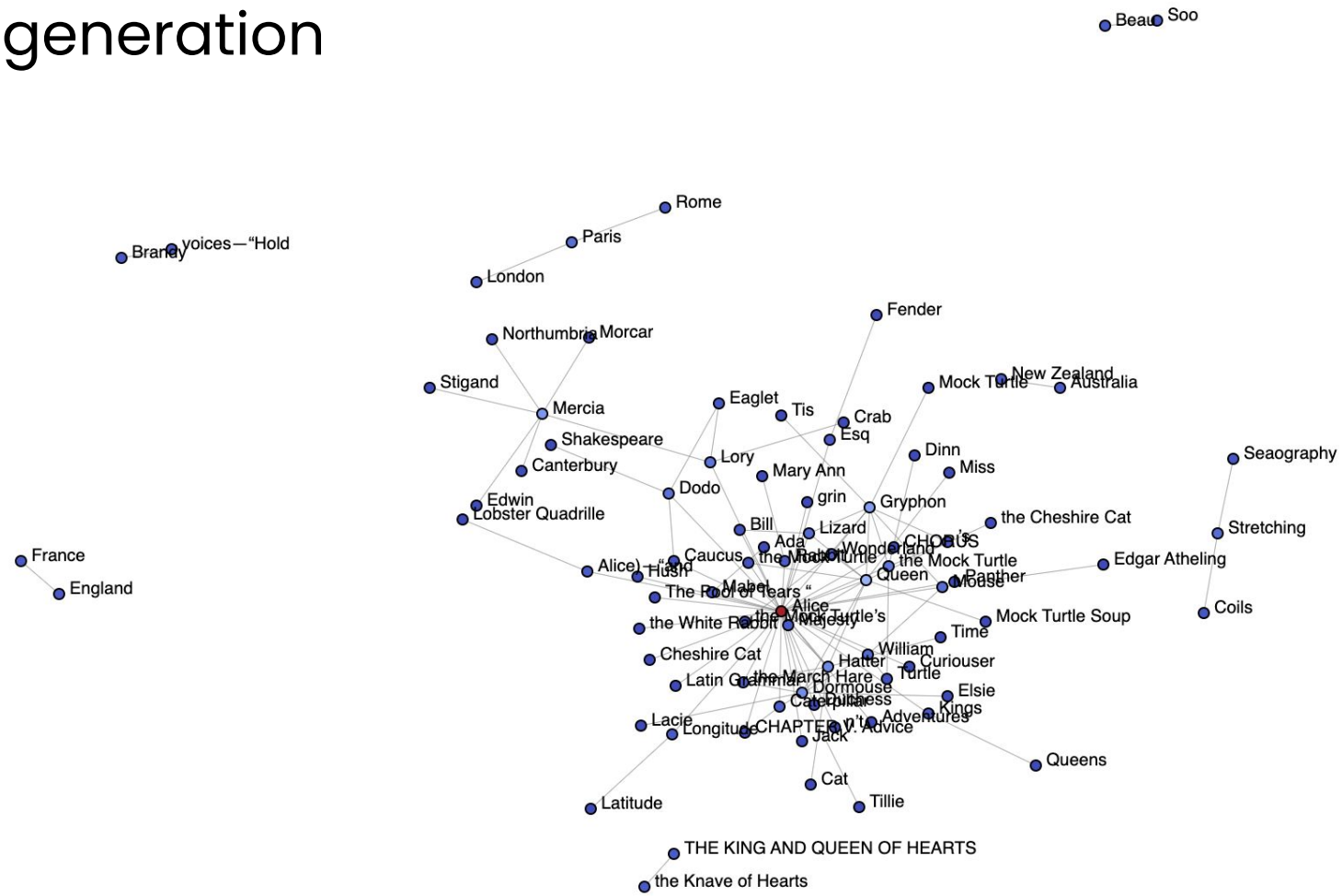
```
alice_entities = extract_entities(text)
alice_entities[0:10]
...
[['Rabbit', 'Alice'],
 ['Longitude', 'Latitude', 'Alice'],
 ['Australia', 'New Zealand'],
 ['the White Rabbit', 'Alice'],
 ['Alice', 'Curiouser', 'The Pool of Tears      “'],
 ['Esq', 'Fender', 'Alice'],
 ['Mabel', 'Ada'],
 ['Paris', 'Rome', 'London'],
 ['Mabel', 'Alice'],
 ['Alice', 'Latin Grammar']]
```

	source	target
0	Rabbit	Alice
1	Longitude	Latitude
2	Longitude	Alice
3	Australia	New Zealand
4	the White Rabbit	Alice

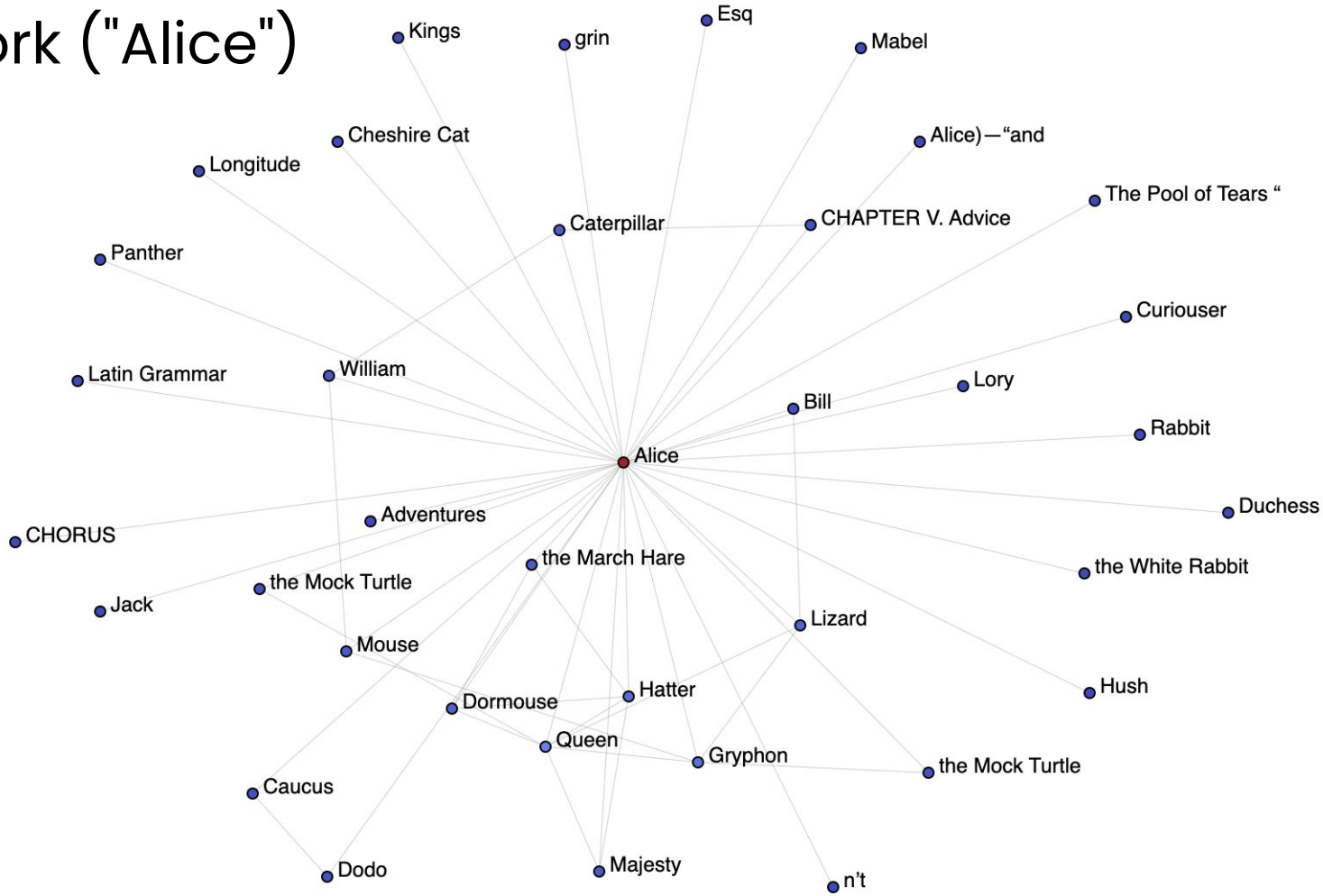
```
alice_entities = extract_entities(text)
alice_entities[0:10]
```

```
...
[['Rabbit', 'Alice'],
 ['Longitude', 'Latitude', 'Alice'],
 ['Australia', 'New Zealand'],
 ['the White Rabbit', 'Alice'],
 ['Alice', 'Curiouser', 'The Pool of Tears      “'],
 ['Esq', 'Fender', 'Alice'],
 ['Mabel', 'Ada'],
 ['Paris', 'Rome', 'London'],
 ['Mabel', 'Alice'],
 ['Alice', 'Latin Grammar']]
```

Network generation



Ego Network ("Alice")



Descrição

O trabalho consiste na realização de uma análise de redes baseada em processamento de linguagem natural (NLP) e ferramentas de grafos. O objetivo é explorar relações linguísticas e criar uma solução completa, indo da análise de texto até a colocação em produção de um grafo interativo, com documentação detalhada em um artigo no **Medium**.

https://medium.com/

Requisito 1

Seleção e Preparação dos Textos

- Escolher um ou mais textos de diferentes fontes (por exemplo, jornais, livros, ou artigos online).
- Caso opte por fontes jornalísticas, comparar grafos gerados de uma mesma notícia veiculada por diferentes meios (Globo, UOL, Carta Capital, etc.).
- Realizar limpeza dos dados (remover linhas desnecessárias, corrigir pontuação, normalizar textos).

Requisito 2

Análise de PoS Tagging e NER

- Usar a biblioteca **NLTK** ou **spaCy** para identificar categorias gramaticais (PoS) e entidades nomeadas (NER).
- Para as categorias gramaticais trabalhar com **NNP.**
- Trabalhar com entidades como **PERSON**, **ORG**, e **GPE**.
- Documentar o processo de análise e salvar os resultados intermediários para inclusão da nota técnica final.

Requisito 3

Geração de Redes

- Criar uma rede com base nas relações entre as entidades extraídas.
- Utilizar a biblioteca NetworkX para construir e manipular a estrutura do grafo.
- Comparar os grafos gerados a partir de diferentes textos ou fontes.

Requisito 4

Análise da Rede

- Calcular métricas como grau, centralidade e densidade.
- Identificar padrões e características, como clusters, hubs ou comunidades relevantes.
- Incorporar visualizações intermediárias (ego network, k-core, etc) para análise qualitativa.

Requisito 5

Visualização e Produção do Grafo:

- Utilizar NetworkX, Gephi para criar uma visualização interativa.
- Colocar o grafo em produção e disponibilizá-lo online, seguindo os métodos apresentados em sala.

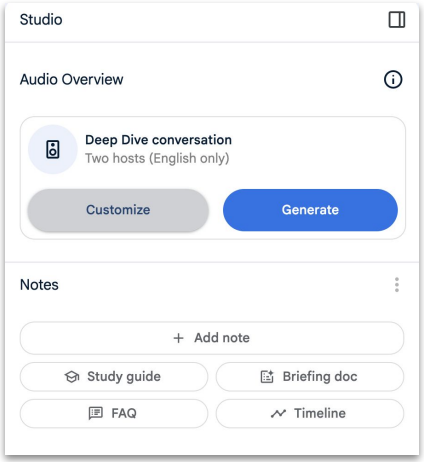
Requisito 6

Documentação e Divulgação:

- Criar um artigo no **Medium** descrevendo detalhadamente todas as etapas realizadas, desde a escolha dos textos até a análise e produção do grafo. Considere esse link como referência.
- O artigo deverá conter:
  - Descrição do processo: objetivos, ferramentas utilizadas e desafios enfrentados.
  - Resultados obtidos: insights sobre os grafos gerados e as análises realizadas.
  - Links relevantes:
    - Link para o repositório GitHub do projeto com o código e os dados utilizados.
    - Link para um podcast, criado com o Notebooklm, explicando os principais conceitos abordados no trabalho.
    - Link para o grafo em produção.
- Submissão no SIGAA: enviar o link do artigo no Medium.

Na geração do podcast: Na função "customize" use um prompt para gerar o podcast em português do Brasil. O áudio do podcast deverá estar na nota técnica no Medium.

Sugestão de prompt: "PT-BR - Crie um podcast em português do Brasil, com dois apresentadores, um homem e uma mulher. O ÁUDIO e o IDIOMA devem ser gerados em PORTUGUÊS do BRASIL para que os brasileiros possam entender. Os apresentadores devem ter um perfil dinâmico, descontraído e interagem entre si."



https://notebooklm.google.com

- Equipe: Trabalho individual ou em dupla.
- Entrega:
  - Artigo publicado no Medium com os requisitos acima.
  - Código organizado no GitHub, com README explicativo e instruções de execução.

Critérios de Avaliação

1. Qualidade do Código (30%):
   - Estrutura, funcionalidade e documentação do código no GitHub.
2. Análise e Produção do Grafo (30%):
   - Relevância das métricas calculadas e insights gerados.
   - Colocação em produção do grafo interativo.
3. Artigo no Medium (20%):
   - Clareza, detalhamento e inclusão dos links necessários.
   - Condição necessária para a nota. Ou seja, sem o artigo a nota do trabalho será zero.
4. Podcast (10%):
   - Explicação clara e coerente dos conceitos, com linguagem dinâmica e acessível.
5. Originalidade e Rigor (10%):
   - Escolha criativa dos textos e rigor na análise realizada.

Janeiro 2025

Se	Te	Qu	Qu	Se	Sá	Do
		1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30	31		

Prazo para submissão: 25 de janeiro às 23h59.

Prova final 29/01