Projeto em Teoria da Computação – MC859
Coleta de dados

Prof. Dr. Ruben Interian

Instituto de Computação, UNICAMP

Resumo

Resumo

Objetivo

- Estudar técnicas de coleta de dados a serem usadas nos seus projetos.

Resumo

Coleta de dados

Antes da coleta de dados, pense em **quais dados você realmente precisa coletar**, e quais podem ser complementados com dados existentes ou disponíveis.

- Existem **datasets** disponíveis publicamente que podem ser essenciais para o seu projeto ou enriquecer as análises a serem realizadas.

Web scraping

Web scraping – extração manual ou automatizada de grandes volumes de dados de sites ou plataformas na Internet.

Web scraping

Web scraping – extração manual ou automatizada de grandes volumes de dados de sites ou plataformas na Internet.

Técnicas de *Web scraping*:

- **Copiar-e-colar**;
- Processamento de páginas **HTML**;
- *Web scraping* usando **APIs**.

Resumo

Web scraping: Wikipedia

Wikipedia – foi criada para ser completamente livre, gratuita e acessível a todos.

Web scraping: Wikipedia

Wikipedia – foi criada para ser completamente livre, gratuita e acessível a todos.

A informação pode ser recuperada:

- Em formato **HTML**;
- Por meio de uma **API**.

Web scraping: Wikipedia

Wikipedia – foi criada para ser completamente livre, gratuita e acessível a todos.

A informação pode ser recuperada:

- Em formato **HTML**;
- Por meio de uma **API**.

Exemplo de uso: obter dados sobre os **Cientistas da Computação** da Wikipedia.

Podemos usar esses dados, por exemplo, para criar uma **rede de interação** entre eles.

Web scraping: Wikipedia

```
# import carrega os módulos que vamos usar
import urllib.request
from bs4 import BeautifulSoup

# URL da página
url = "https://en.wikipedia.org/wiki/Alan_Turing"

# Fazemos a requisição e leitura dos dados
source = urllib.request.urlopen(url).read()
soup = BeautifulSoup(source, 'lxml')

# A função find_all faz a busca pelos parágrafos do texto
paragraphs = []
for paragraph in soup.find_all('p'):
    paragraphs.append(str(paragraph.text))
text = '\n'.join(paragraphs) # Juntamos tudo

print(text[:1000])
```

executed in 533ms

```
Alan Mathison Turing OBE FRS (/ˈtjʊərɪŋ/; 23 June 1912 – 7 June 1954) was an English mathematician, computer scientist, logicia
n, cryptanalyst, philosopher and theoretical biologist.[5] He was highly influential in the development of theoretical computer
science, providing a formalisation of the concepts of algorithm and computation with the Turing machine, which can be considere
d a model of a general-purpose computer.[6][7][8] Turing is widely considered to be the father of theoretical computer science.
[9]

Born in London, Turing was raised in southern England. He graduated in maths from King's College, Cambridge, and in 1938, earne
d a maths PhD from Princeton University. During the Second World War, Turing worked for the Government Code and Cypher School a
t Bletchley Park, Britain's codebreaking centre that produced Ultra intelligence. He led Hut 8, the section responsible for Ger
```

Web scraping: Wikipedia

Principais pacotes usados:

- `requests` – permite fazer solicitações HTTP.

Web scraping: Wikipedia

Principais pacotes usados:

- `requests` – permite fazer solicitações HTTP.
- `BeautifulSoup` – torna mais fácil pesquisar e extrair informação de páginas web em HTML.

Web scraping: Wikipedia (API)

Além dessa forma 'tradicional' de fazer *Web scraping*, a Wikipedia possui uma **API**: interface de programação de aplicações.

Uma **API** permite que um *software* possa se comunicar com outro. Se os dados podem ser acessados por meio de uma **API**, geralmente é melhor usar ela!

Exemplo de consulta à **API**: $\longrightarrow$ API request

Web scraping: Wikipedia (API)

```
# import carrega os módulos que vamos usar
import urllib.request, json

# URL do endpoint (recurso) da API
url = "https://en.wikipedia.org/w/api.php?action=query&prop=extracts&explaintext&titles={}&format=json"

# Fazemos a requisição e leitura dos dados
response = urllib.request.urlopen(url.format("Alan_Turing"))
data = json.load(response) # data contem todo o artigo

text = list(data['query']['pages'].values())[0]['extract']
print(text[:1000])
```

executed in 517ms

```
Alan Mathison Turing  (; 23 June 1912 – 7 June 1954) was an English mathematician, computer scientist, logician, cryptanalyst,
philosopher and theoretical biologist. He was highly influential in the development of theoretical computer science, providing
a formalisation of the concepts of algorithm and computation with the Turing machine, which can be considered a model of a gene
ral-purpose computer. Turing is widely considered to be the father of theoretical computer science.
Born in London, Turing was raised in southern England. He graduated in maths from King's College, Cambridge, and in 1938, earne
d a maths PhD from Princeton University. During the Second World War, Turing worked for the Government Code and Cypher School a
t Bletchley Park, Britain's codebreaking centre that produced Ultra intelligence. He led Hut 8, the section responsible for Ger
man naval cryptanalysis. Turing devised techniques for speeding the breaking of German ciphers, including improvements to the p
```

Web scraping: Wikipedia (API)

Para **mais informações** sobre:

- API,
- Formato dos dados,
- Tutorial e documentação,
- Exemplos mais complexos,

link: $\longrightarrow$ Central de pesquisas – Portal de dados – API

Web scraping: Wikipedia (API)

Como podemos criar a **rede** dos cientistas?

- Acessar a página: $\longrightarrow$ List of computer scientists

Web scraping: Wikipedia (API)

Como podemos criar a **rede** dos cientistas?

- Acessar a página: $\longrightarrow$ List of computer scientists
- Extrair automaticamente a **lista de links** para páginas dos cientistas.

Web scraping: Wikipedia (API)

Como podemos criar a **rede** dos cientistas?

- Acessar a página: $\longrightarrow$ List of computer scientists
- Extrair automaticamente a **lista de links** para páginas dos cientistas.
- Extrair **cada página** usando *Web scraping*, como mostrado nos slides.

Web scraping: Wikipedia (API)

Como podemos criar a **rede** dos cientistas?

- Acessar a página: $\longrightarrow$ List of computer scientists
- Extrair automaticamente a **lista de links** para páginas dos cientistas.
- Extrair **cada página** usando *Web scraping*, como mostrado nos slides.
- Construir a **rede**, identificando os cientistas que aparecem nas páginas de outros cientistas.

Resumo

Pacotes para *Web scraping*: `snscrape`

Pacote `snscrape` para Python.

Instruções: `https://github.com/JustAnotherArchivist/snscrape`

Pacotes para *Web scraping*: `snscrape`

Pacote `snscrape` para Python.

Instruções: `https://github.com/JustAnotherArchivist/snscrape`

- Lista de plataformas: Facebook, Instagram, Reddit, Telegram, ~~Twitter~~, Weibo (declaradamente).

Pacotes para *Web scraping*: `snscrape`

Pacote `snscrape` para Python.

Instruções: `https://github.com/JustAnotherArchivist/snscrape`

- Lista de plataformas: Facebook, Instagram, Reddit, Telegram, ~~Twitter~~, Weibo (declaradamente).
- Dependendo da plataforma, `snscrape` permite recuperar: usuários e perfis, grupos ou comunidades, hashtags, canais, listar posts.

Pacotes para *Web scraping*: `snscrape` – Exemplo de uso

```python
import snscrape.modules.telegram as sntel
for i, post in enumerate(sntel.TelegramChannelScraper('TelegramTipsBR').get_items()):
    if i == 3:
        break
    print(post.url)
    print(post.date)
    print(post.content)
    print(post.outlinks)
```

executed in 1.12s

```
https://t.me/s/TelegramTipsBR/334
2024-07-26 14:13:46+00:00
⭐ Widget de Link nos Stories. Usuários Premium podem destacar um link no story adicionando um widget estiloso – com várias opç
ões de layout e cores.Widgets de link, tags de localização e mais podem ser adicionados ao seu story a partir da guia de sticke
rs no editor de stories.
['https://t.me/TelegramTipsBR/333', 'https://t.me/TelegramTipsBR/269']
https://t.me/s/TelegramTipsBR/333
2024-07-25 22:42:50+00:00
Busque Stories por Localização. Usuários de férias ou em um evento podem adicionar uma tag de localização ao story para mostrar
onde estavam.Tocar em uma tag de localização abre outros stories públicos do mesmo lugar – para ver mais selfies de shows ou re
comendações de restaurantes.Apenas stories que o usuário tem permissão para ver aparecem nos resultados de busca. Stories priva
dos nunca são incluídos em buscas públicas.
[]
https://t.me/s/TelegramTipsBR/332
2024-07-22 20:18:45+00:00
Busque Stories por Hashtag. Tocar em hashtags nas legendas dos stories permite navegar pelos stories públicos com hashtags corr
espondentes.Se você é um criador ou uma empresa, pode usar hashtags para aumentar seu público de forma orgânica e alcançar novo
s clientes.
['https://t.me/TelegramTipsBR/327']
```

Pacotes para *Web scraping*: `linkedin-api`

Pacote `linkedin-api`

Instruções: `https://github.com/tomquirk/linkedin-api`

Pacotes para *Web scraping*: `linkedin-api`

Pacote `linkedin-api`

Instruções: `https://github.com/tomquirk/linkedin-api`

- Requer uma conta válida no LinkedIn.

Pacotes para *Web scraping*: `linkedin-api`

Pacote `linkedin-api`

Instruções: `https://github.com/tomquirk/linkedin-api`

- Requer uma conta válida no LinkedIn.
- Recomendação: usar com precaução, pois não é uma API "oficial" do LinkedIn. **Não tente fazer milhões de consultas desde a sua conta!**

Pacotes para *Web scraping*: `linkedin-api` – Usuários

```
import os, yaml
import pandas as pd
from linkedin_api import Linkedin

config_data = yaml.safe_load(open(os.path.join(os.path.dirname("__file__"), "auth/auth.yaml"), "r"))
api = Linkedin(config_data['user'], config_data['pass'])
profile = api.get_profile('rubeninterian')
profile
```

executed in 7.22s

```
{'summary': 'I am a Computer Science researcher studying human interaction networks.',
 'industryName': 'Technology, Information and Internet',
 'lastName': 'Interian',
 'locationName': 'Brazil',
 'student': False,
```

```
profile = api.get_profile('williamhgates')
profile
```

executed in 3.32s

```
{'summary': 'Co-chair of the Bill & Melinda Gates Foundation.
 'industryName': 'Philanthropic Fundraising Services',
 'lastName': 'Gates',
 'locationName': 'United States',
 'student': False,
```

Pacotes para *Web scraping*: `linkedin-api` – Contatos

```
connections = api.get_profile_connections('rubeninterian')
df_connections = pd.DataFrame(connections)
print(f"Número de conexões: {len(df_connections)}")
df_connections.drop(columns='name', inplace=True)
df_connections.head()
```

executed in 2m 29s

```
Número de conexões: 377
```

	urn_id	distance	jobtitle	location
0	ACoAABf9sNcB1WYFWFTIPwLNJYk36uGsfgr5hSk	DISTANCE_1	Researcher on Dimensional and Computational Me...	São Paulo, Brazil
1	ACoAAD1uZl8BsYiigekHYvtO8zi3XKeg2693tlo	DISTANCE_1	Senior Backend Developer	Brasília, DF
2	ACoAABwiGcUB0iaH8k7JRQzakQKjMF4BslbH0YQ	DISTANCE_1	CTO SocialHub | Signal & Image Processing | Vi...	Rio de Janeiro, Brazil
3	ACoAADN0jJwB6f-bCkm5AnXf-oqent0obVaTY9M	DISTANCE_1	BsC, MsC, Biochemistry, UH/UNU-BIOLAC'S Felows...	Ribeirão Preto, SP
4	ACoAAB4wI-kBwPuUK4kv8o6jmQt1oHeGgtn2Wiw	DISTANCE_1	PhD in Computing	Niterói, RJ

Pacotes para *Web scraping*: `linkedin-api` – Outros métodos

Outros métodos no `linkedin-api`:

- `get_company`, `get_company_updates`
- `get_job`, `get_feed_posts`, `get_post_comments`
- `get_profile_network_info`, `get_profile_posts`, `get_profile_skills`
- `search`, `search_companies`, `search_jobs`, `search_people`

Pacotes para *Web scraping*: `g1-news-scraping`

Web scraping em sites de noticias. **Exemplo**: `g1-news-scraping` para o site **G1**.

Instruções: `https://github.com/leviobrabo/G1-news-scraping`

Pacotes para *Web scraping*: `g1-news-scraping`

Web scraping em sites de noticias. **Exemplo**: `g1-news-scraping` para o site **G1**.

Instruções: `https://github.com/leviobrabo/G1-news-scraping`

- É um dos scrapers mais simples. Se quiser aprender como fazer, veja o código!

Pacotes para *Web scraping*: `g1-news-scraping`

Web scraping em sites de noticias. **Exemplo**: `g1-news-scraping` para o site **G1**.

Instruções: `https://github.com/leviobrabo/G1-news-scraping`

- É um dos scrapers mais simples. Se quiser aprender como fazer, veja o código!
- Retorna uma lista das últimas notícias publicadas no G1, em ordem cronológica inversa (primeiro as noticias mais novas).

Pacotes para *Web scraping*: g1-news-scraping

```
import pandas as pd
from scraperg1 import get_news
noticias = get_news(3)
df = pd.DataFrame(noticias)
print(f"Número de noticias: {len(df)}")
df = df[['title', 'autor', 'link', 'full_text']]
df.head()
```

executed in 3.07s

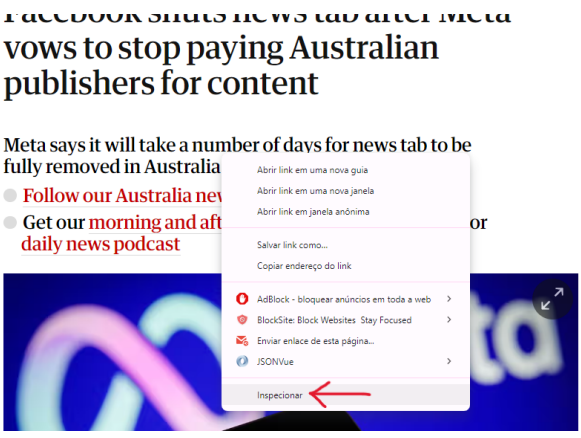
```
INFO:scraperg1:Obtendo notícias...
INFO:scraperg1:Notícia recebida
INFO:scraperg1:Notícia recebida
INFO:scraperg1:Notícia recebida
INFO:scraperg1:3 notícias obtidas.
```

```
Número de noticias: 3
```

	title	autor	link	full_text
0	'Animais olímpicos': veja espécies que são des...	Por Fernanda Machado, Terra da Gente	https://g1.globo.com/sp/campinas-regiao/terra-...	Depois do judô, o atletismo é o esporte que ma...
1	Detran-BA faz leilão de carros em 15 cidades; ...	Por g1 BA	https://g1.globo.com/ba/bahia/noticia/2024/08/...	O Departamento Estadual de Trânsito da Bahia (...
2	Painel do "Amazônia Que Eu Quero" será transmi...	Por Leonardo Matheus, Fundação Rede Amazônica...	https://g1.globo.com/ac/acre/noticia/2024/08/0...	Nesta próxima quarta-feira (07), o painel "Bio...

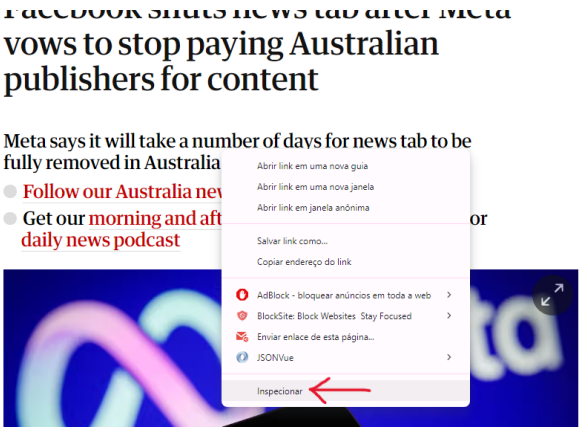
Outras ferramentas: "Inspecionar"

Função **Inspecionar** do Chrome:



Outras ferramentas: "Inspecionar"

Função **Inspecionar** do Chrome:



$\Rightarrow$ Browser.

Outras ferramentas: WhatsApp Explorer

WhatsApp Explorer: ferramenta criada para coletar dados do WhatsApp, com consentimento explícito do usuário, de forma anônima, e focando em grupos com pelo menos 4 pessoas.

Documento e instruções:
WhatsApp Explorer: A Data Donation Tool To Facilitate Research on WhatsApp

Ferramenta: $\longrightarrow$ whatsapp.whats-viral.me

Pacotes para *Web scraping*

Exemplo BONUS: `pyktok` para TikTok.

Instruções: `https://github.com/dfreelon/pyktok`

Resumo

Repositórios e datasets

Repositórios e datasetes: verifique se existe o dataset com os dados que você precisa.

Repositórios e datasets

Repositórios e datasetes: verifique se existe o dataset com os dados que você precisa.

- **Stanford Large Network Dataset Collection**
  Coleção de datasets com diferentes tipos de redes.
  URL: `https://snap.stanford.edu/data/`

Repositórios e datasets

Repositórios e datasetes: verifique se existe o dataset com os dados que você precisa.

- **Stanford Large Network Dataset Collection**
  Coleção de datasets com diferentes tipos de redes.
  URL: `https://snap.stanford.edu/data/`
- **Kaggle**
  Plataforma e comunidade de usuários online que publica **datasets**
  URL: `https://www.kaggle.com/datasets`

Repositórios e datasets

ConvoKit

Repositório com ferramentas e datasets, principalmente de conversas e diálogos.

URL: `https://convokit.cornell.edu/`

URL: `https://github.com/CornellNLP/ConvoKit`

Repositórios e datasets

ConvoKit
Repositório com ferramentas e datasets, principalmente de conversas e diálogos.
URL: `https://convokit.cornell.edu/`
URL: `https://github.com/CornellNLP/ConvoKit`

- **Diálogos** em filmes;

Repositórios e datasets

ConvoKit
Repositório com ferramentas e datasets, principalmente de conversas e diálogos.
URL: `https://convokit.cornell.edu/`
URL: `https://github.com/CornellNLP/ConvoKit`

- **Diálogos** em filmes;
- Conversas no **Reddit**;

Repositórios e datasets

ConvoKit
Repositório com ferramentas e datasets, principalmente de conversas e diálogos.
URL: `https://convokit.cornell.edu/`
URL: `https://github.com/CornellNLP/ConvoKit`

- **Diálogos** em filmes;
- Conversas no **Reddit**;
- Todas as **conversas** das 10 temporadas da série **Friends**;

Repositórios e datasets

ConvoKit
Repositório com ferramentas e datasets, principalmente de conversas e diálogos.
URL: `https://convokit.cornell.edu/`
URL: `https://github.com/CornellNLP/ConvoKit`

- **Diálogos** em filmes;
- Conversas no **Reddit**;
- Todas as **conversas** das 10 temporadas da série **Friends**;
- **Entrevistas** a jogadores de **tênis**;
- **E muitos outros**!

Repositórios e datasets

Datasets específicos:

- **Publicações e citações – Scopus**. Periodicidade: poucos dias.
  URL: `https://www.scopus.com/search/form.uri`
  Permite recuperar a rede de colaborações e citações!

Repositórios e datasets

Datasets específicos:

- **Publicações e citações – Scopus**. Periodicidade: poucos dias.
  URL: `https://www.scopus.com/search/form.uri`
  Permite recuperar a rede de colaborações e citações!
- **Empresas no Brasil**. Periodicidade: mensal. Atualização: 13/07/2024.
  URL: `https://dados.gov.br/dados/conjuntos-dados/cadastro-nacional-da-pessoa-juridica---cnpj`
  Mais recursos: `https://github.com/turicas/socios-brasil`
  Permite recuperar os sócios de cada empresa!

Repositórios e datasets

Datasets específicos:

- **Publicações e citações – Scopus**. Periodicidade: poucos dias.
  URL: `https://www.scopus.com/search/form.uri`
  Permite recuperar a rede de colaborações e citações!
- **Empresas no Brasil**. Periodicidade: mensal. Atualização: 13/07/2024.
  URL: `https://dados.gov.br/dados/conjuntos-dados/cadastro-nacional-da-pessoa-juridica---cnpj`
  Mais recursos: `https://github.com/turicas/socios-brasil`
  Permite recuperar os sócios de cada empresa!
- **Interaction Web DataBase** – datasets com redes de interação de espécies hospedado pelo Departamento de Ecologia da USP.
  URL: `http://www.ecologia.ib.usp.br/iwdb/resources.html`

Repositórios e datasets

Existem muitos outros datasets e repositórios!

Coleta de dados

Observação crucial: **NUNCA** coletar ou extrair informações que são **identificáveis**.

Coleta de dados

Observação crucial: **NUNCA** coletar ou extrair informações que são **identificáveis**.

Exemplos:

1. Não coletar o nome completo.

Coleta de dados

Observação crucial: **NUNCA** coletar ou extrair informações que são **identificáveis**.

Exemplos:

1. Não coletar o nome completo.
2. Não coletar CPF, número de documento de identidade, CNH.

Coleta de dados

Observação crucial: **NUNCA** coletar ou extrair informações que são **identificáveis**.

Exemplos:

1. Não coletar o nome completo.
2. Não coletar CPF, número de documento de identidade, CNH.
3. Não coletar endereço de PF (pode coletar o estado, ou no máximo a cidade).

Coleta de dados

Observação crucial: **NUNCA** coletar ou extrair informações que são **identificáveis**.

Exemplos:

1. Não coletar o nome completo.
2. Não coletar CPF, número de documento de identidade, CNH.
3. Não coletar endereço de PF (pode coletar o estado, ou no máximo a cidade).
4. Não coletar fotos ou alguma outra informação que permita identificar o indivíduo.

Dúvidas

Dúvidas?