Explorando consultas SPARQL na Wikidata com Python: Tipificação de Metadados e Reconciliação de Dados

Luis Felipe Rosa de Oliveira

Dalton Lopes Martins

UnB
Programa de Pós-graduação
Ciência da Informação - PPGCINF
FACULDADE DE CIÊNCIA DA INFORMAÇÃO

INTRODUÇÃO

- Contexto:
  - Laboratório de Inteligência de Redes
  - Projeto Tainacan (MARTINS; CARVALHO JÚNIOR; GERMANI, 2019, p.60)

- Temática:
  - Web Semântica
  - Dados Abertos Ligados

- Reconciliação de Dados:
  - Processo permite **reconhecer entidades**, seja no formato textual ou no formato de objeto digital em uma ou mais **bases de conhecimento on-line**, permitindo assim **formar conexões entre bases** por meio de identificadores únicos. (SANDERSON, 2016)

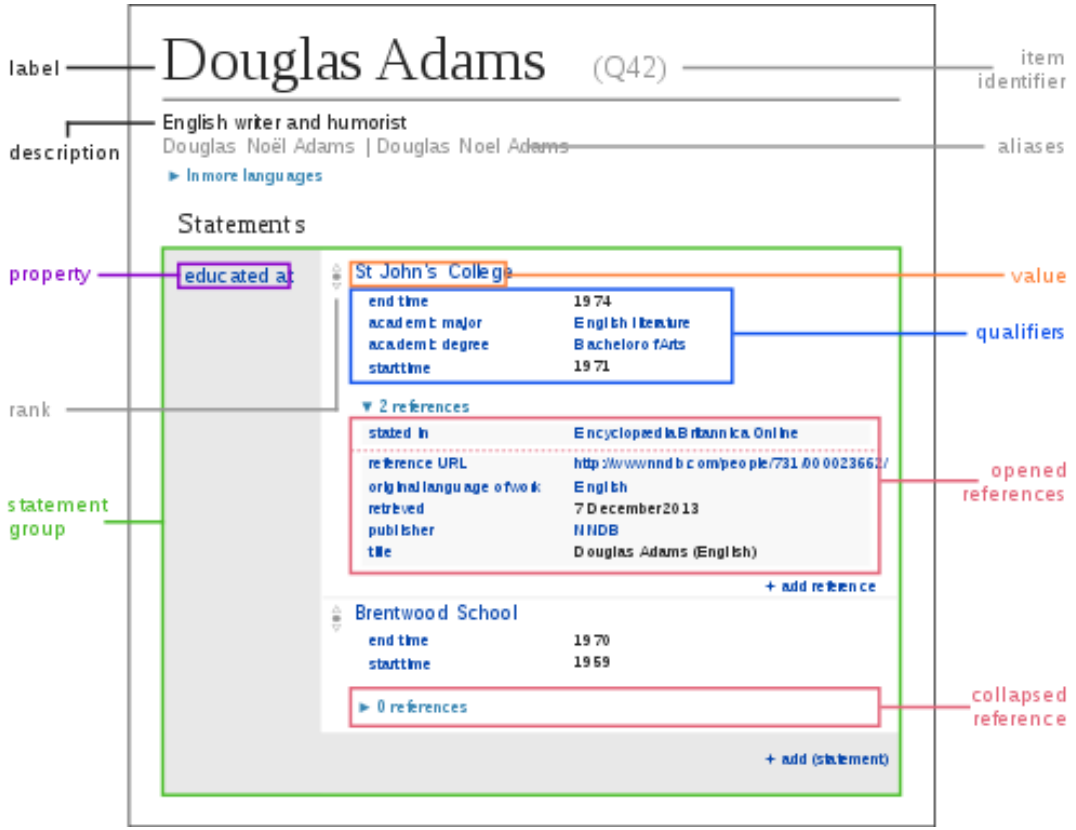
INTRODUÇÃO

A Wikidata (Base de Conhecimento)

“é uma **base de conhecimento livre** e aberta que pode ser lida e editada por humanos e máquinas” WIKIDATA (2019)

Como ligar meus dados com a Wikidata?

Como desenvolver processos semiautomáticos de reconciliação de dados com a Wikidata?



WIKIDATA

INTRODUÇÃO

Proposta da Pesquisa

- Transformar objetos textuais em objetos digitais (ZENG, 2019)
- *Linked Open Data*

Desenvolvimento de Dois Scripts:

- Tipificação de metadados, com o objetivo de reconhecer a qual instância da Wikidata determinado metadado pertence a partir de seu conjunto de valores.

- Reconciliação de dados, em que determinados valores textuais são buscados na Wikidata em busca de identificar seus possíveis objetos representativos.

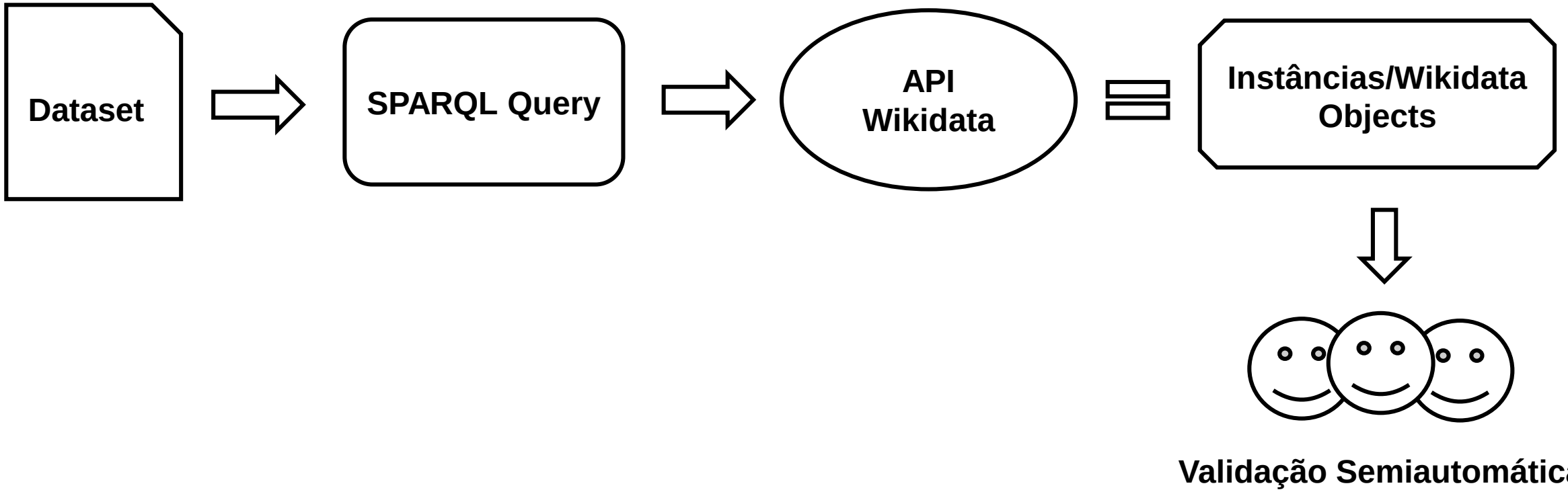
INTRODUÇÃO

O que se espera:

- Compreender melhor **como dados de diferentes acervos podem ser ligados de forma semiautomática**, complementando propostas de formas mais diretas de se **reconhecer entidades em bases de conhecimento, e enriquecer bases de dados locais**.

Método

Processo de Reconciliação



Método

Python

“é uma linguagem de programação que te permite trabalhar rapidamente e integrar sistemas de forma mais efetiva.”
(PYTHON 2019)

Bibliotecas:

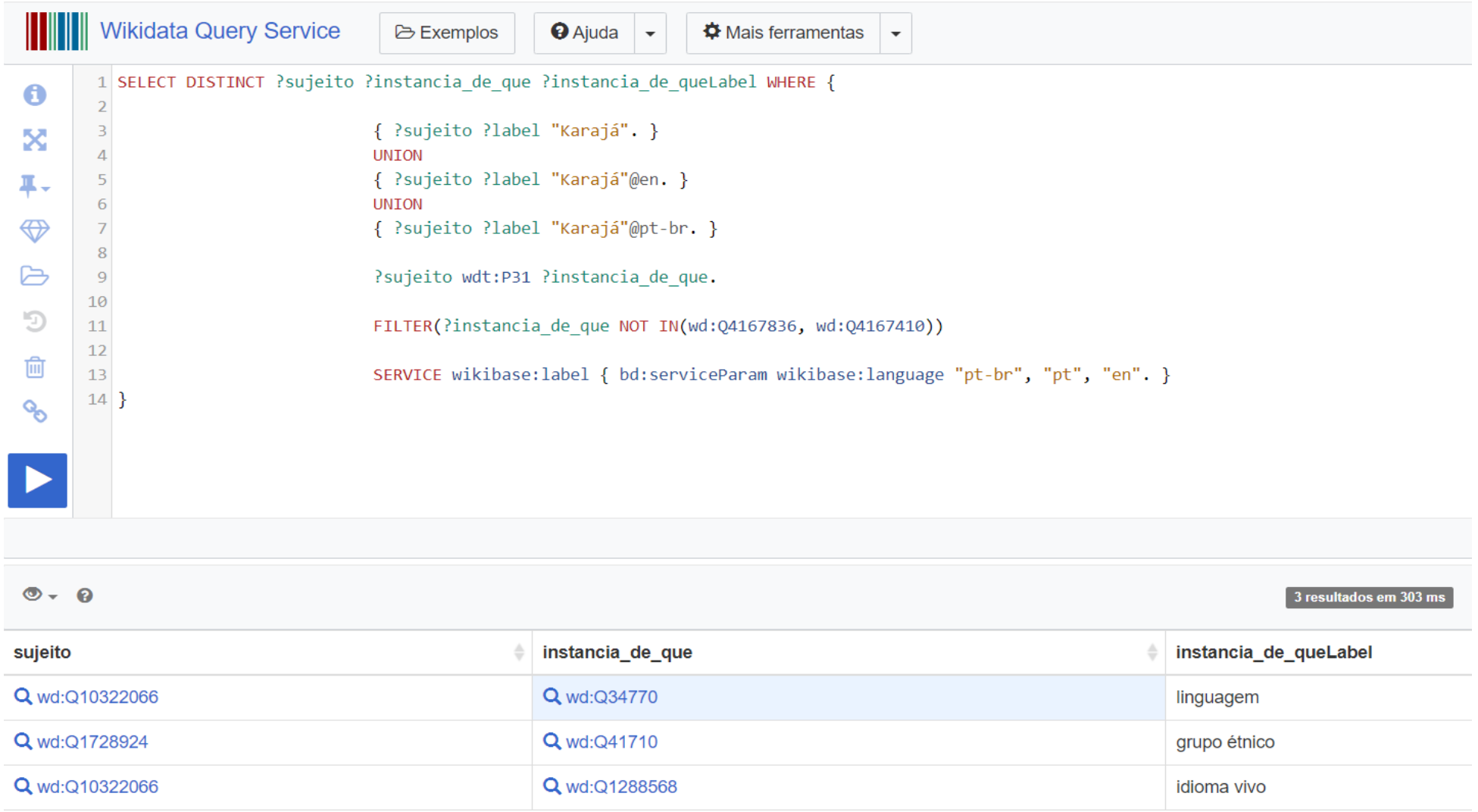
- Requests
- Pandas
- Time
- Fuzzywuzzy

Método

Dataset

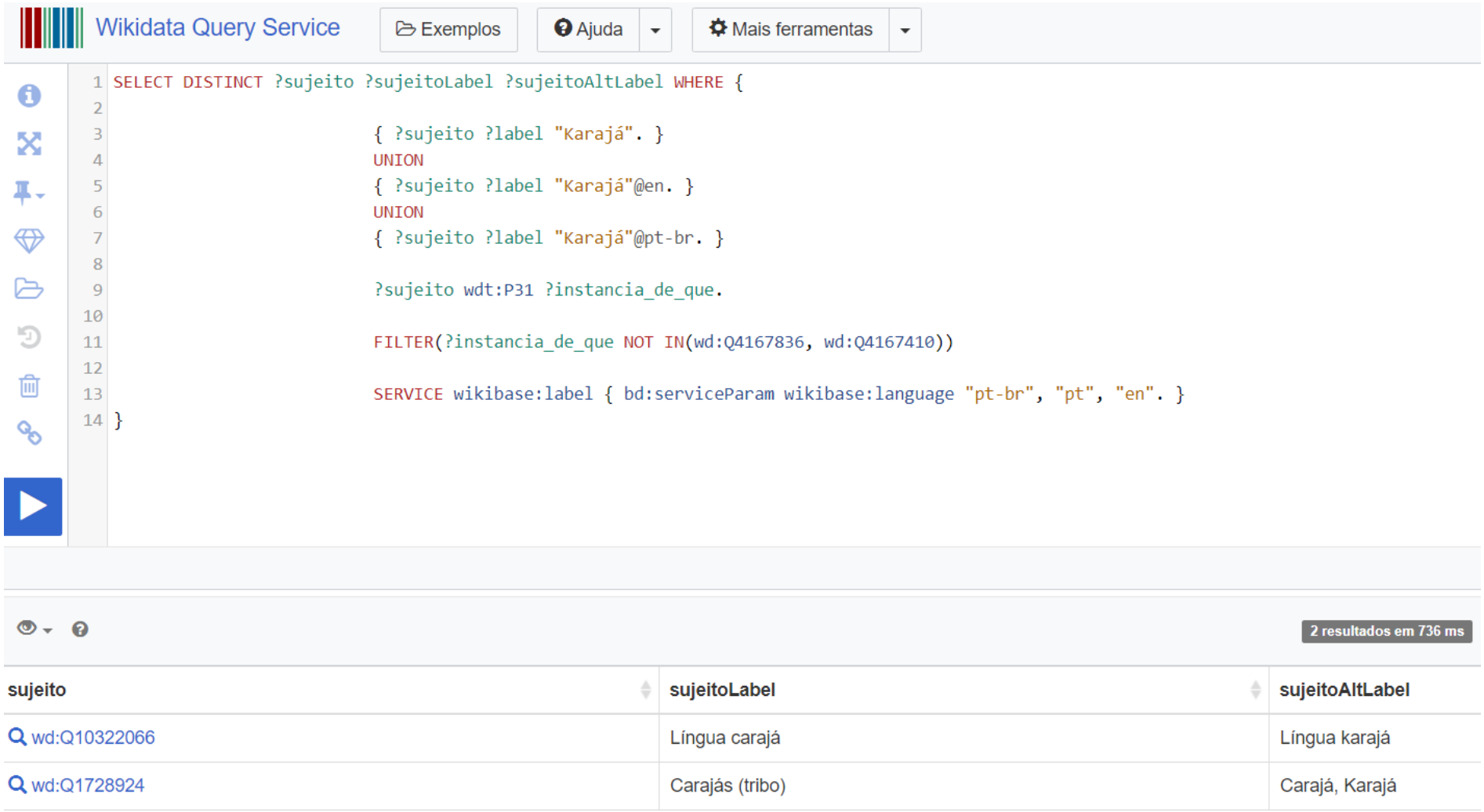
Cidade	País	Marcas
Goiânia	Brasil	Coca-Cola
São Paulo	Chile	Arno
Brumadinho	França	Epson
Planaltina	Estados U	Dell
Curitiba	Butão	C&A
São Carlos	Argentina	CaBas
Abaetetuba	Marrocos	Calvin Klein

Método (SPARQL Query – Buscando Instâncias)



sujeito	instancia_de_que	instancia_de_queLabel
wd:Q10322066	wd:Q34770	linguagem
wd:Q1728924	wd:Q41710	grupo étnico
wd:Q10322066	wd:Q1288568	idioma vivo

Método (SPARQL Query – Buscando Objetos)



sujeito	sujeitoLabel	sujeitoAltLabel
wd:Q10322066	Língua carajá	Língua karajá
wd:Q1728924	Carajás (tribo)	Carajá, Karajá

Método (Script – Tipificação de Metadados)

```
# Script to map metadata to wikidata entities (instance of)
#Required Packages
import requests
import pandas as pd
import time

#API Endpoint
api_url = 'https://query.wikidata.org/sparql'
#Values to reconcile database
tabela_df = pd.read_csv('planilha_teste.csv', encoding='utf-8')
```

Método (Script – Tipificação de Metadados)

```
#For each metadata/column on values to reconcile database
for column in database.columns:
    print("Trabalhando no metadado ", column)

    #For each value of a column/metadata
    for value in tabela_df[column]:
        print("Procurando Resultados para ", value)

        #SPARQL query. Returns subject QID, instance QID and instance label, searching the value for subjects
        #with matching labels.
        query = """ SELECT DISTINCT ?sujeito ?instancia_de_que ?instancia_de_queLabel WHERE {
                      { ?sujeito ?label "%s". }
                      UNION
                      { ?sujeito ?label "%s"@en. }
                      UNION
                      { ?sujeito ?label "%s"@pt-br. }
                      ?sujeito wdt:P31 ?instancia_de_que.
                      FILTER(?instancia_de_que NOT IN(wd:Q4167836, wd:Q4167410))
                      SERVICE wikibase:label { bd:serviceParam wikibase:language "pt-br", "pt", "en". }
                    }""" %(value, value, value)

        #Request the query to wikidata api and outputs a json.
        r = requests.get(api_url, params = {'format': 'json', 'query': query})
        data = r.json()
```


```
#Request the query to wikidata api and outputs a json.
r = requests.get(api_url, params = {'format': 'json', 'query': query})
data = r.json()
```

Método (Script – Tipificação de Metadados)

```
#Dataframe to store results
result_df = pd.DataFrame(columns = ['column','value','qid','qid_instance','instance_label'])


  #For each result inset data on result_df
  for item in data['results']['bindings']:
      result_df = result_df.append({'column':column,'value':value,
                                    'qid':item['sujeito']['value'],
                                    'qid_instancia':item['instancia']['value'],
                                    'instance_label':item['instanciaLabel']['value'] },
                                     ignore_index=True)
  time.sleep(1)
```

```
print(reconcile_database(tabela_df).to_csv("resultado.csv", encoding='utf-8'))
```

Resultado (Script - Tipificação de Metadados)

column	value	qid	qid_instance	instance_label
Cidade	Goiânia	http://www.wikidata.org/entity/Q83189	http://www.wikidata.org/entity/Q515	cidade
Cidade	Goiânia	http://www.wikidata.org/entity/Q83189	http://www.wikidata.org/entity/Q5119	capital
País	Brasil	http://www.wikidata.org/entity/Q155	http://www.wikidata.org/entity/Q859563	Estado secular
País	Brasil	http://www.wikidata.org/entity/Q10344998	http://www.wikidata.org/entity/Q1761072	parque estadual
País	Brasil	http://www.wikidata.org/entity/Q4957796	http://www.wikidata.org/entity/Q7366	canção
País	Brasil	http://www.wikidata.org/entity/Q1799348	http://www.wikidata.org/entity/Q3184121	município do Brasil
País	Brasil	http://www.wikidata.org/entity/Q155	http://www.wikidata.org/entity/Q3624078	estado soberano
País	Brasil	http://www.wikidata.org/entity/Q20274905	http://www.wikidata.org/entity/Q486972	povoado
Marcas	Epson	http://www.wikidata.org/entity/Q159614	http://www.wikidata.org/entity/Q4830453	empresa
Marcas	Epson	http://www.wikidata.org/entity/Q159614	http://www.wikidata.org/entity/Q6881511	empresa
Marcas	Epson	http://www.wikidata.org/entity/Q37054258	http://www.wikidata.org/entity/Q101352	sobrenome

Método (Script – Reconciliação de Dados)

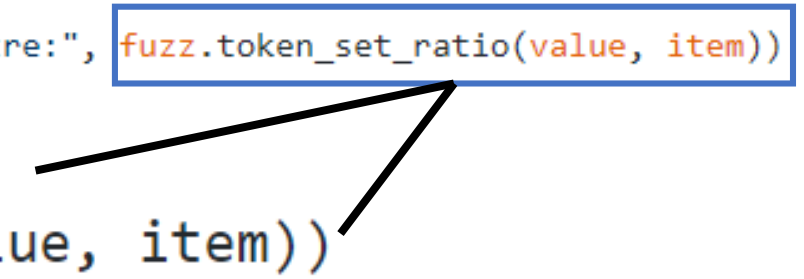
```
#Required Packages
import requests
import pandas as pd
import time
from collections import defaultdict
from fuzzywuzzy import fuzz
labels_dict = defaultdict(list)

#API Endpoint
api_url = 'https://query.wikidata.org/sparql'

#Values to reconcile database
tabela_df = pd.read_csv('planilha_teste.csv', encoding='utf-8')

value = 'Karajá'
```

Método (Script – Reconciliação de Dados)

```
query = """ SELECT DISTINCT ?sujeito ?sujeitoLabel ?sujeitoAltLabel WHERE {

                            { ?sujeito ?label "%s". }
                            UNION
                            { ?sujeito ?label "%s"@en. }
                            UNION
                            { ?sujeito ?label "%s"@pt-br. }

                            ?sujeito wdt:P31 ?instance.

                            SERVICE wikibase:label { bd:serviceParam wikibase:language "pt-br", "pt", "en". }
                          }"""%(value, value, value)

r = requests.get(api_url, params = {'format': 'json', 'query': query})
data = r.json()
```

Método (Script – Reconciliação de Dados)

```
#Faz um dicionário com lista das labels para cada resultado retornado
for item in data['results']['bindings']:
    try:
        labels_dict[item['sujeito']['value']] = item['sujeitoAltLabel']['value'].split(',')
        labels_dict[item['sujeito']['value']].append(item['sujeitoLabel']['value'])
    except:
        labels_dict[item['sujeito']['value']].append(item['sujeitoLabel']['value'])

#Limpar Repetições da lista de labels
for key in labels_dict.keys():
    labels_dict[key] = list(set(labels_dict[key]))

#Avaliar o score do match entre o valor procurado com o valor procurado
for key in labels_dict.keys():
    for item in labels_dict[key]:
        print("Valor:", value, "|| Objeto", key, "|| Label:", item, "|| Socre:", fuzz.token_set_ratio(value, item))
```



Resultado (Script – Reconciliação de Dados)

```
Valor: Karajá || Objeto http://www.wikidata.org/entity/Q10322066 || Label: Língua karajá || Score: 100
Valor: Karajá || Objeto http://www.wikidata.org/entity/Q10322066 || Label: Língua carajá || Score: 50
Valor: Karajá || Objeto http://www.wikidata.org/entity/Q1728924 || Label: Carajás (tribo) || Score: 47
Valor: Karajá || Objeto http://www.wikidata.org/entity/Q1728924 || Label: Carajá || Score: 80
Valor: Karajá || Objeto http://www.wikidata.org/entity/Q1728924 || Label:  Karajá || Score: 100
```

Referências

- MARTINS, D. L.; CARVALHO, J. M. C; GERMANI, L. Projeto Tainacan: Experimentos, Aprendizados e Descobertas da Cultura Digital no Universo dos Acervos das Instituições Memoriais. In: **TIC CULTURA Pesquisa Sobre o Uso das Tecnologias de Informação e Comunicação nos Equipamentos Culturais Brasileiros — 2018.** Comitê Gestor da Internet no Brasil, São Paulo, 2019.

- SANDERSON, R. **"The Linked Data Snowball and Why We Need Reconciliation"**, 2016, Disponível em: https://www.slideshare.net/azaroth42/linked-data-snowball-or-why-we-need-reconciliation. Acesso em: 15/08/2019.

- ZENG, M. L. "Semantic enrichment for enhancing LAM data and supporting digital humanities. Review article". **El profesional de la información**, v. 28, n. 1, 2019.

OBRIGADO

Luis Felipe Rosa | luisrosaprof@gmail.com

Dalton Martins | dmartins@gmail.com

Scripts – https://github.com/luisfeliperd/wikidata_sparql