Universidade do Minho

Departamento de Informática

Scripting e Processamento de Linguagem Natural

Trabalho Prático nº 3

Enunciado nº 4

WebScraping + Ontologia

Henrique Ribeiro

PG38415

Introdução

No seguimento da unidade curricular de Scripting no Processamento de Linguagem Natural para o trabalho prático número 3 foi escolhido o enunciado número 4: WebScraping + Ontologia. As metas para este enunciado consistem em desenvolver software capaz de recolher a informação de um website e posteriormente criar uma ontologia representante da informação recolhida.

Foi escolhido o website https://www.lexico.pt/ para a realização deste projeto. Este website consiste em um dicionário da Língua Portuguesa online, nele podemos pesquisar as definições de milhares de palavras através de um layout simples e eficiente.

WebScraping

WebScraping trata-se de uma técnica de coleção de dados de websites através de software com o objetivo de utilizar os dados recolhidos para futuro processamento. Para conseguir um webscraping eficiente deve-se fazer primeiro um estudo do website alvo, neste estudo procura-se saber em que localizações da estrutura HTML os dados estão localizados para poder-se proceder à procura dos mesmos via software.

Para a realização do webscraping foi feito uso de uma biblioteca bastante poderosa denominada “BeautifulSoup” que permite a realização de pedidos e captação de informação devolvida por parte de um website.

Inicia-se o scraping com a inserção de um URL para visita e extração dos dados, após isto recolhem-se os dados relevantes procurando as tags HTML identificadas no estudo realizado como portadoras destes mesmos. Na imagem seguinte está representada a função “getRelations” que tem como objetivo procurar palavras sinónimas e antónimas à palavra alvo. Utiliza-se um objecto “soup” que é obtido através do pedido feito com a biblioteca “BeautifulSoup”, de seguida percorrem-se as tags “div” de classe “card card-pl” encontradas pelo website e caso sejam encontradas as palavras “Sinónimo” ou “Antónimo” dentro dessas tags, adiciona-se a informação à lista correspondente. Após tratamento das strings extraídas e transformação para unicode as listas de palavras sinónimas e antónimas são devolvidas.

```
def getRelations(soup):
    synonyms = []
    antonyms = []
    for relation in soup.find_all('div',class_='card card-pl'):
        if 'Sinónimo' in str(relation.h2):
            synonyms=relation.p.text.split(',')
        if 'Antónimo' in str(relation.h2):
            antonyms=relation.p.text.split(',')

    for index,word in enumerate(synonyms):
        if ' ' in word:
            synonyms[index]=word[1:]
        synonyms[index] = unidecode.unidecode(synonyms[index])

    for index,word in enumerate(antonyms):
        if ' ' in word:
            antonyms[index]=word[1:]
        antonyms[index] = unidecode.unidecode(antonyms[index])

    return synonyms,antonyms
```

Nas duas imagens seguintes está presente a função “getMeanings” que é utilizada para extrair os significados das palavras. Inicia-se com a procura de certas tags e da preparação dos dados para utilização futura, esta preparação passa por colocar cada significado numa posição da lista “meanings” e cada classificação da palavra na lista “wordType”.

```
hadNumbers = 0
for i in range(len(meanings)):
    for j in range(len(meanings[i])):
        if hasNumber(meanings[i][j]):
            hadNumbers=1
            wordType[i] = meanings[i][0]
        meanings[i][j] = re.sub(r'([0-9])\.\s','',meanings[i][j])
        meanings[i][j] = meanings[i][j].replace('\'','``')
        if not meanings[i][j]:
            del meanings[i][j]

#delete meanings copied to wordType
try:
    if(hadNumbers):
        for i in range(len(wordType)):
            del meanings[i][0]
except Exception as e:
    print('Error in getting meaning')
    print(e)

meanings = [ele for ele in meanings if ele != []]
wordType = [ele for ele in wordType if ele != []]


for i,word in enumerate(wordType):
    if 'v.' in word:
        wordType[i] = 'verbo'
    if 'n.' in word:
        wordType[i] = 'nome'
    if 'adj.' in word:
        wordType[i] = 'adjetivo'
    if 'adv.' in word:
        wordType[i] = 'adverbio'
    if 'art.' in word:
        wordType[i] = 'artigo'
    if 'det.' in word:
        wordType[i] = 'determinante'
    if not word:
        wordType[i] = 'vazio'
```

Após a obtenção do significado e classificação da palavra procede-se ao processamento adicional da lista “wordType” com a função “replaceWordTypes”. Esta

função tem como objectivo analisar as classificações obtidas através do scraping e encontrar a classificação correspondente mais próxima entre as opções : adjetivo, advérbio, artigo, conjunção, determinante, interjeição, preposição, pronome, substantivo e verbo. Esta função devolve então as opções que foram correspondidas.

```
def replaceWordTypes(wordTypes,word):
    types =['adjetivo','adverbio','artigo','conjuncao','determinante','interjeicao','preposicao','pronome','substantivo','verbo','nome','vazio' ]
    matches = []

    found = False
    cutoff = 0.4

    while found==False:
        try:
            for type in wordTypes:
                matches.append(get_close_matches(type,types,1,cutoff)[0])
                found = True
        except Exception as e:
            cutoff=cutoff-0.1
            if cutoff < 0.1:
                    found = True
                    matches.append('vazio')
                    print('Cutoff')

    for index,match in enumerate(matches):
        if 'nome' in match:
            matches[index] = 'substantivo'

    return matches
'''
```

Após recolher o significado, classificação, sinónimos e antónimos da palavra é então criada e populada a ontologia.

Ontologia

Uma ontologia envolvida no software development trata-se de uma maneira de representar diversas propriedades e relações de um conceito através da definição de categorias e classificações que o representam.

Para criar uma ontologia do dicionário da Língua Portuguesa definiram-se 2 classes: “Word” e “Type”. Definiram-se também as relações simétricas “eSinonimo”, “eAntonimo” e “temTipo”. Estas relações tornam a ontologia numa estrutura de dados poderosa que permite navegar facilmente as palavras existentes nela mesma. Por fim foi definida a propriedade “Significado” para colocar o significado extraído das palavras.

```
#################################################################
    #    Data properties
    #################################################################

:Significado rdf:type owl:DatatypeProperty ;
             rdfs:domain :Word .

#################################################################
    #    Object Properties
    #################################################################

###  http://www.tartesdajulia.com/ontologies/2020/5/untitled-ontology-58#eSinonimo
    :eSinonimo rdf:type owl:ObjectProperty ,
                        owl:SymmetricProperty .

###  http://www.tartesdajulia.com/ontologies/2020/5/untitled-ontology-58#eAntonimo
    :eAntonimo rdf:type owl:ObjectProperty ,
                        owl:SymmetricProperty .

:temTipo rdf:type owl:ObjectProperty ;
         rdfs:domain :Word ;
         rdfs:range :Type .
```

```
#################################################################
    #    Classes
    #################################################################

    :Type rdf:type owl:Class ;
      owl:equivalentClass [ rdf:type owl:Class ;
                            owl:oneOf ( :adjetivo
                                        :adverbio
                                        :artigo
                                        :conjucao
                                        :determinante
                                        :interjeicao
                                        :preposicao
                                        :pronome
                                        :substantivo
                                        :verbo
                                      )
                          ] .

:Word rdf:type owl:Class .
```

Após a criação da estrutura da ontologia procede-se à população com as palavras obtidas. Para isto fez-se uso de uma função “createIndividual” que cria as strings e imprime corretamente para o ficheiro desejado, concluindo assim a ontologia.

```
def createIndividual(mainWord,types,meanings,synonyms,antonyms,ontologyName):
    file = codecs.open('DicionarioLECTO.ttl','a+','utf-8')

    header = '\n\n:' + mainWord.replace(' ','-') +' rdf:type owl:NamedIndividual ,\n\t\t\t:Word ;'

    type = ''
    for i in range(len(types)):
        type += '\n\t\t:temTipo :' + types[i] +';'

    meaning = ''
    for i in range(len(types)):
        for j in range(len(meanings[i])):
            meaning +=':Significado \''+meanings[i][j]+'\' ;'
    raw_s = r'{}'.format(meaning)
    antonym = ''
    for ant in antonyms :
        antonym += '\n\t\t:eAntonimo :' + ant +';'

    synonym = ''
    for syn in synonyms :
        synonym += '\n\t\t:eSinonimo :' + syn +';'

    file.write(header)
    file.write(type)
    meaning =repr(meaning)[1:-1]
    file.write('\n\t\t'+meaning)
    file.write(antonym)
    file.write(synonym)
    file.write('.')
    file.close
```

Resultados

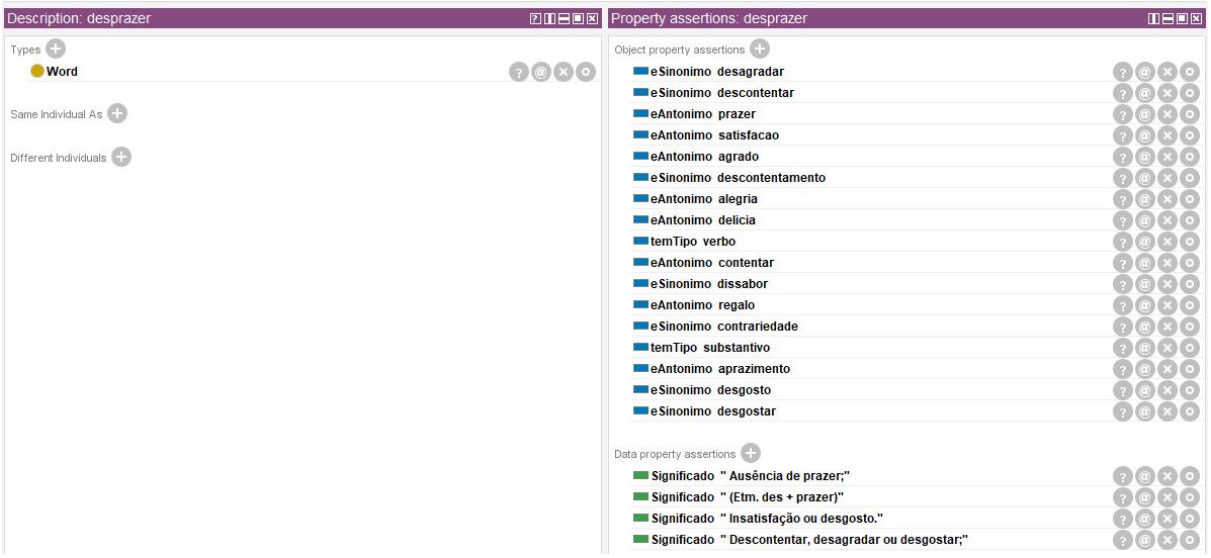
Ao executar o programa denominado de “scraper.py” este irá percorrer as letras do alfabeto em pesquisa de palavras no website escolhido imprimindo cada uma para o ficheiro da ontologia que irá conter uma lista de tamanho vasto de indivíduos (palavras recolhidas)

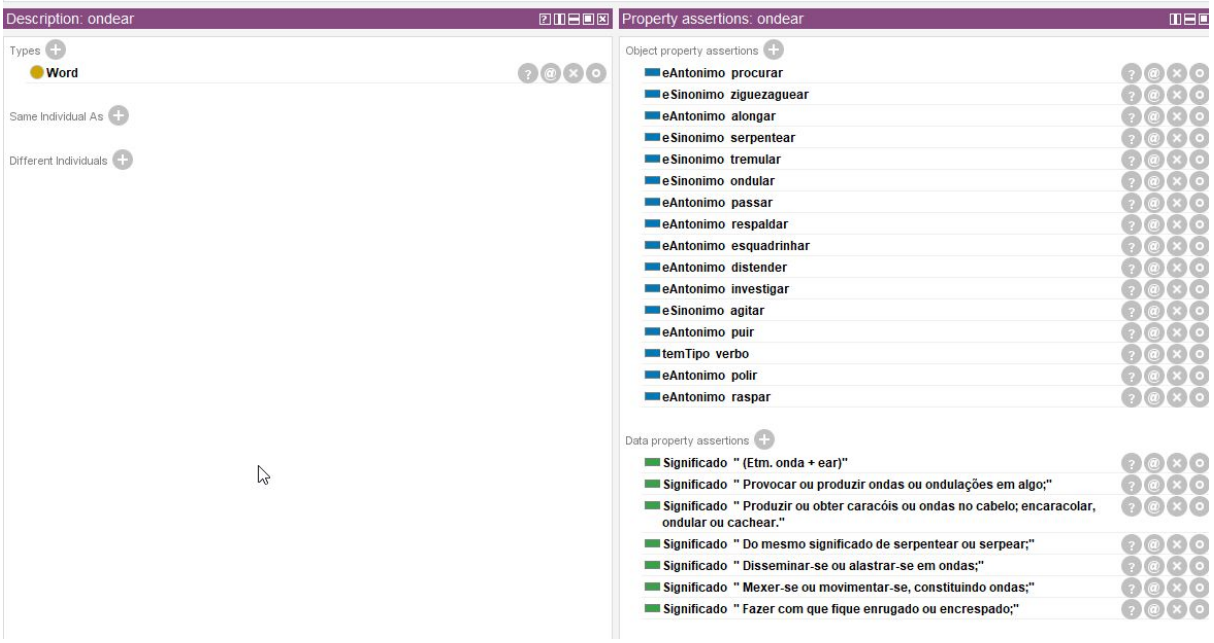
```
alphabet = list(string.ascii_lowercase)

for letter in alphabet:
    wordList = [letter]
    getWords(wordList)
```

Após carregamento para o GraphDB e feita uma query de contagem de número de palavras temos o resultante de 15938 palavras como mostra a imagem.

	nPalavra
1	"15938"^^xsd:integer

Como exemplo de palavras inseridas apresentam-se as imagens:




Description: ondear
Types
Word
Same Individual As
Different Individuals
Property assertions: ondear
Object property assertions
eAntonimo procurar
eSinonimo ziguezaguear
eAntonimo alongar
eSinonimo serpentear
eSinonimo tremular
eSinonimo ondular
eAntonimo passar
eAntonimo respaldar
eAntonimo esquadrinhar
eAntonimo distender
eAntonimo investigar
eSinonimo agitar
eAntonimo puir
temTipo verbo
eAntonimo polir
eAntonimo raspar
Data property assertions
Significado " (Etm. onda + ear)"
Significado " Provocar ou produzir ondas ou ondulações em algo;"
Significado " Produzir ou obter caracóis ou ondas no cabelo; encaracolar, ondular ou cachear."
Significado " Do mesmo significado de serpentear ou serpear;"
Significado " Disseminar-se ou alastrar-se em ondas;"
Significado " Mexer-se ou movimentar-se, constituindo ondas;"
Significado " Fazer com que fique enrugado ou encrespado;"