Scripting no Processamento de Língua Natural

Teste, 2022-06-23

1 Python

1. O texto abaixo corresponde ao conteúdo de um ficheiro chamado "nomes-adjectivos.txt".

```
Europa      Europeu
Portugal     Português
França      Francês
Itália      Italiano
Austrália   Australiano
...
```

Cada linha deste ficheiro contêm duas palavras (nomes e correspondente adjetivo), separadas por um tab. Escreva uma função em python que, dado um texto como input, converte todos os adjetivos de nacionalidade nos seus nomes correspondentes. A função deve devolver o texto convertido.

2. Rescreva a função seguinte substituindo o ciclo aninhado por uma lista em compreensão aninhada:

```
words = ['attribution', 'confabulation', 'elocution',
        'sequoia', 'tenacious', 'unidirectional']
vsequences = set()
for word in words:
    vowels = []
    for char in word:
        if char in 'aeiou':
            vowels.append(char)
    vsequences.add(''.join(vowels))
sorted(vsequences)

#['aiuio', 'eaiou', 'eouio', 'euoia', 'oauaio', 'uiieioa']
```

2 Spacy

1. Crie um programa em python que calcule e imprima todos os verbos contidos num ficheiro de texto.
2. Calcule o número de ocorrências dos lemma dos verbos calculados anteriormente e guarde o resultado num ficheiro JSON.

3 BS4

No exemplo abaixo estão apresentados excertos de páginas html de um website que disponibiliza livros de medicina. Implemente um programa em python, usando BS4, que dadas essas páginas faz download de todos os livros disponibilizados por esse website.

Página com categorias dos livros: https://www.infolivros.org/medicina

```
...
<nav class="tableofcontents" role="navigation" aria-label="Tabela de Conteúdos">
  <div class="table-of-content-wrap">
    <div class="table-of-contents-title-wrap">
        <span class="table-of-contents-title">Tabela de Conteúdos</span>
    </div>
    <ul class="table-of-content-list table-cont">
        <li>1. <a class="contents__entry" href="/medicina/anatomia">Livros de Anatomia</a></li>
```

```
    </ul>
    <ul class="table-of-content-list table-cont">
        <li>2. <a class="contents__entry" href="/medicina/cardiologia">Livros de Cardiologia</a></li>
    </ul>
    <ul class="table-of-content-list table-cont">
        <li>3. <a class="contents__entry" href="/medicina/cirurgia">Livros de Cirurgia</a></li>
    </ul>
    <ul class="table-of-content-list table-cont">
        <li>4. <a class="contents__entry" href="/medicina/clinica-medica">Livros de Clínica Médica</a><
    </ul>
    <ul class="table-of-content-list table-cont">
        <li>5. <a class="contents__entry" href="/medicina/dermatologia">Livros de Dermatologia</a></li>
    </ul>
    <ul class="table-of-content-list table-cont">
        <li>6. <a class="contents__entry" href="/medicina/diabetes">Livros de Diabetes</a></li>
    </ul>
    ...

  </div>
</nav>
```

Página com livros da categoria (https://www.infolivros.org/medicina/clinica-medica)

```
...
<div class="query-pdf-container">

   <div class="Livros_Container">
      <div class="Livros_Texto">
         <h3 class="Livros_Titulo">1) O guia do Jovem Internista</h3>
         <p class="Livros_Atribucion">Gustavo Carvalho, Maria Antunes, Cristina Forte</p>
         Fonte: <a href="https://cssjd.org.br/O-guia-do-Jovem-Internista.pdf" target="_blank">
         <span style="color:#545454">Complexo de Saúde São João de Deus</span></a>
         <p class="Livros_Atribuicao"></p>
      </div>
      <div class="Livros_Botoes">
      <a href="/pdfview/O-guia-do-Jovem-Internista.pdf" class="Botao_um" target="_blank">Ler</a>
      <a href="/download/O-guia-do-Jovem-Internista.pdf" class="Botao_dois" target="_blank">Baixar</a><
   </div>
   <div class="Livros_Container">
      <div class="Livros_Texto">
         <h3 class="Livros_Titulo">2) Manual de Insuficiência Cardíaca</h3>
         <p class="Livros_Atribuicao">Complexo de Saúde São João de Deus</p>
         Fonte: <a href="https://cssjd.org.br/Manual-de-Insuficiência-Cardíaca.pdf" target="_blank">
         <span style="color:#545454">Complexo de Saúde São João de Deus</span></a>
         <p class="Libros_Atribuicao"></p>
      </div>
      <div class="Livros_Botoes">
      <a href="/pdfview/Manual-de-Insuficiência-Cardíaca.pdf" class="Botao_um" target="_blank">Ler</a>
      <a href="/download/Manual-de-Insuficiência-Cardíaca.pdf" class="Botao_2" target="_blank">Baixar</
   </div>
  ...
</div>
```

4 Expressões Regulares, Anotador de Entidades

Dispomos de um dicionário contendo milhões de nomes de entidades associadas ao respectivo tipo (resultado de uma travessia da Wikipedia). Todas estas entidades: - começam com maiúscula - podem ter de, do, da, dos, das em palavra interior

```
DE={ "Miranda do Douro": "cidade", "Platão": "filósofo", "Violeta Parra": "cantautor", ... }
```

Escreva uma função que dado um texto, procure uma as entidades, e se existirem no dicionário DE, as anote com seu tipo, e as mantenha inalteradas em caso contrário:

```
Consta que Platão estudou em Miranda so Douro e Constantim
   ↓
Consta que <e t="filósofo">Platão</e> estudou em <e t="cidade">Miranda do Douro</e> e Constantim
```

5 Espaços em falta

Após uma operação de OCR, um determinado texto perdeu completamente os espaços. Pretendemos construir uma ferramenta que reponha os espaços em falta. Suponha ainda que:

- Dispomos ainda de um dicionário pt (`freq[pal]` → frequência-relativa de pal)
- As palavras têm dimensão menor que 15 caracteres.

```
                    1                   2                   3                   4
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2
a c a v a l o d a d o , n ã o s e o l h a o d e n t e D i t a d o s F N A C 1 9 9 9 .
-   -   -   -   -   -       -   - -     - -   -     -       -   -       -    w1 (len=1)
  ----------- --- ---         ---           ---   ---         ---            w6 w2
        -----           -----       ----------                                w3
  -------     -------   -------   -------     ------- -------       -------  w4
    -------                                 --------- -------------           w4 w5 w7
```

1. Crie uma função que construa o grafo de palavras(GP). O GP é um dicionário (início → (fim+1, pal)*) ou seja, a chave é o índice início de palavra e o valor associado é uma lista de triplos (fim+1, palavra, freq)

```
gp = mkgp("acavalodado")
{ 0 : [ (1, a, 13.5) ]
  1 : [ (5, cava, 0.1), (7, cavalo, 3.2)]
  2 : [ (3, a, 13.5), (6, aval, 1.5)]
  4 : [ (5, a, 13.5), (7, alo, 0.5)]
  6 : [ (7, o, 13,2) ]
  7 : [ (9, da, 12.1), (11, dado,4)]
  9 : [ (11, do, 10.7) ]
  10: [ (11, o, 13,2) ]
}
```

2. Crie uma função que dado um GP e uma posição inicial, e uma posição terminal, calcule uma frase válida, dando prioridade às palavras mais longas. `frase(gp, 0 , 11)` poderia dar "a cavalo dado"
3. Diga informalmente como poderia usar as frequências para obter uma frase mais optimizada.

6 Directory Walk (os.walk)

Para guardar um arquivo, optou-se por criar uma ontologia dispersa em árvore de diretorias. Cada classe X, fica pasta "C-X". Sub-classes ficam em sub-pastas. Cada individuo Y de uma class Z, fica uma pasta "I-Y" (que vai conter os ficheiros a ele associados), Esta pasta fica dentro da pasta C-Z.

Segue um exemplo de uma árvore de diretorias arquivo, e em comentário, notas acerca da ontologia associada (classes, indivíduos e triplos).

```
- A-Eurico_Tomás_Lima     # arquivo: Eurico Tomás de Lima || triplos
  - C-doc                 # class: doc       || (doc, a, class)
    - C-foto              # class: foto      || (foto, a, class) (foto, isSubclass, doc)
    - C-carta             # class: carta     || (carta, a, class) (carta, isSubclass, doc)
      - I-c1              # individuo: c1    || (c1, a, carta)
        - meta-c1.yaml    # meta de c1       || (c1, meta, "meta-c1.yaml")
        - img-c1.jpg      #                  || (c1, img, "img-c1.jpg")
      - I-c2              # ... outro individuo carta
        ....
      - C-postal          # (postal, a, class) (postal, isSubclass, carta)
    - C-partitura
```

1. Dado uma diretoria arquivo (Ex: A-Eurico_Tomás_Lima) imprima todos os indivíduos nela presentes e sua classe ou seja os pares (Individuo, class).
2. Crie uma script makeindiv foto-c20 que procure a pasta "C-voto", crie a pasta I-c20, e arranque com o editor meta-c20.yaml
3. Dado uma diretoria arquivo devolva o conjunto dos triplos a ela associados `triplos("A-Eurico_Tomás_Lima")` devolve `[(doc, a, class) (carta, a, class) (carta, isSubclass, doc) (c1 a carta) ...]`

Sugestão: Considere a função `os.walk` que dada uma diretoria, visita todas as suas subdiretoria

```
os.walk = walk(dirraiz, topdown=True)
    Directory tree generator.

    Procura todas as diretorias nessa árvore de subdiretoria, (incluindo a raiz)
    e devolve (yields) uma lista de triplos (dirpath, dirnames, filenames)

       dirpath is a string, the path to the directory.
       dirnames is a list of the names of the subdirectories in dirpath.
       filenames is a list of the names of the non-directory files in dirpath.

    Note that the names in the lists are just names, with no path components.
    To get a full path (which begins with top) to a file or directory in
    dirpath, do os.path.join(dirpath, name).

    Se o argumento 'topdown' for true ou não especificado, visita topdown senão: bottom-up

Uso típico:
    for dir, dirs, files in os.walk("/home/jj/spln"):
       print(f'Pasta {dir} contem: {len(dirs)} pastas e {len(files)} ficheiros')
```