INFORMÁTICA MÉDICA

Trabalho Prático 1

BEATRIZ RODRIGUES, PG53696
BRUNO MACHADO, PG53709
RÚBEN GANANÇA, PG54203

DOCUMENTOS

Glossário
Ministério Saúde

Glossário
Termos Médicos

WIPO Pearl
Covid-19 Glossary

Anatomia
Geral

Objetivo.

siglas.json

abreviacoes.json

wipo.json

conceitos.json

+

termos_glosario.json

+

glossario_anatomia.json

=

glossario_geral.json

Documento Glossário Ministério Saúde

Etapa

- Imports e leitura do ficheiro;

```
import re, json

filename = "glossario_ministerio_saude.xml"

with open(filename, 'r', encoding='utf-8') as file:
    ficheiro = file.read()
```

Documento Glossário Ministério Saúde

Etapa

- Remoção de informação não relevante, manualmente e através do Python;
- Remoção de linhas do xml que dificultassem a extração da informação;
- Criação de marcas (início da categoria com “*”).

```
#Remocao de linhas não necessárias
ficheiro = re.sub(r"</?page.*?>",r"",ficheiro) #serve para remover a tag page
ficheiro = re.sub(r"</?pdf2xml.*?>",r"",ficheiro)
ficheiro = re.sub(r"</?image.*?>",r"",ficheiro)

padrao = r'<text.*font="(22|15|25|23)".*>.*</text>\n?' #padrao para remover tudo o que for texto com font="15" ou "22" ...
padrao2 = r'<text.*top="238".*>.*</text>\n?' #Remove texto dos cabeçalhos
ficheiro = re.sub(padrao, r"", ficheiro)
ficheiro = re.sub(padrao2, r"", ficheiro)
ficheiro = re.sub (r"</?text.*?>", "", ficheiro) #*? para ele parar no primeiro > e não retirar info importante
ficheiro = re.sub (r"<i>Categoria: </i>", "*", ficheiro)
ficheiro = re.sub (r"- ", "", ficheiro) #usado para tirar quando é quebra de linha
ficheiro = re.sub (r" □ ", "-", ficheiro)
ficheiro = re.sub('^\s*$', r"", ficheiro, flags=re.MULTILINE) #Remove linhas vazias
ficheiro = re.sub (r"<i>", "", ficheiro) #É preciso ser retirado isto porque há termos em ingles em italico como "Western blot"
ficheiro = re.sub (r"</i>", "", ficheiro)
```

Documento Glossário Ministério Saúde

Etapa

- Grupo de captura para juntar linhas consecutivas do nome do conceito;

- Grupo de captura para rreter a informação do conceito;

```
ficheiro = re.sub(r'</b>\n<b>(.*?)</b>\n', r'\1</b>\n', ficheiro) #Junta duas linhas consecutivas de <b> ou seja de nome de termos cortados

lista = re.findall(r"<b>(.*?)</b>\n(?:\*\n)?(.*?)\n(.*?)<b>", ficheiro, re.DOTALL)
```

Documento Glossário Ministério Saúde

Etapa

- Tratamento dos grupos de captura;
- Inserção dos conceitos num dicionário e posteriormente num json.

```
# Processar os termos
novos_conceitos = []
glossario = {}
for termo, categoria, descricao in lista:
    novo_termo = termo.strip()
    novo_termo = re.sub (r"<b>", "", novo_termo)
    novo_termo = re.sub (r"</b>", "", novo_termo)
    novo_termo = re.sub (r"\n", "", novo_termo)
    nova_categoria = categoria.strip()
    nova_categoria = re.sub (r"<b>", "", nova_categoria)
    nova_categoria = re.sub (r"</b>", "", nova_categoria)
    nova_categoria = re.sub (r"\n", "", nova_categoria)
    nova_descricao = descricao.strip()
    nova_descricao = re.sub (r"\n", "", nova_descricao)
    if novo_termo not in glossario:
        if nova_descricao !="":
            glossario[novo_termo] = {"Categoria": nova_categoria, "Descricao": nova_descricao}
        elif nova_descricao == "":
            glossario[novo_termo] = {"Categoria": "Sem Categoria", "Descricao": nova_categoria}

file_out = open("conceitos.json","w",encoding= 'utf-8')
json.dump(glossario,file_out,indent=4,ensure_ascii=False)
file_out.close()
```

Documento Glossário Ministério Saúde

Json de conceitos obtido


```
{
    "Abordagem médica tradicional do adulto hospitalizado": {
        "Categoria": "Atenção à Saúde",
        "Descricao": "Focada em uma queixa principal e o hábito médico de tentar explicar todas as queixas e os sinais por um único
        diagnóstico, que é adequada no adulto jovem-não se aplica em relação ao idoso."
    },
    "Abuso incestuoso": {
        "Categoria": "Acidentes e Violência",
        "Descricao": "Consiste no abuso sexual envolvendo pais ou outro parente próximo, os quais se encon-tram em uma posição de maior
        poder em re-lação à vítima."
    },
    "Abuso sexual na infância": {
        "Categoria": "Acidentes e Violência",
        "Descricao": "É todo ato ou jogo sexual, relação heterosse-xual ou homossexual, cujo agressor está em estágio de desenvolvimento
        psicossexual mais adiantado que a criança ou adolescente. Tem por intenção estimulá-la sexualmente ou uti-lizá-la para obter
        satisfação sexual. Essas prá-ticas eróticas e sexuais são impostas à criança ou adolescente pela violência física, por amea-ças
        ou pela indução de sua vontade."
    },
```

Documento Glossário Ministério Saúde

Json de siglas

```
siglas = re.findall(r"<b>(.+?)</b>\n([^<]+)",ficheiro) #separa a sigla da sua descricao

# Processar as siglas
novos_conceitos = []
for designacao,descricao in siglas:
    nova_desig = designacao.strip()
    nova_desig = re.sub (r"\n", "", nova_desig)
    nova_descri = descricao.strip()
    nova_descri = re.sub (r"\n", "", nova_descri)
    novos_conceitos.append((nova_desig,nova_descri))


conceitos_dict = dict(novos_conceitos)

file_out = open("siglas.json","w",encoding= 'utf-8')
json.dump(conceitos_dict,file_out,indent=4,ensure_ascii=False)
file_out.close()
```

Documento Glossário Ministério Saúde

Json de siglas


```
{
    "AB": "Atenção Básica",
    "ABEn": "Associação Brasileira de Enfermagem",
    "ADT": "Assistência Domiciliar Terapêutica",
    "AFE": "Autorização de Funcionamento de Empresa",
    "AIDPI": "Atenção Integrada às Doenças Prevalentes na Infância",
    "AIDS": "Síndrome da Imunodefi ciência Adquirida",
    "AIH": "Autorização de Internação Hospitalar",
    "AIS": "Ações Integradas de Saúde",
```

Documento Glossário Termos Médicos

Etapa

- Imports e leitura do ficheiro;

```
import re
import json

filename = "GlossaÌ▯rio de Termos MeÌ▯dicos TeÌ▯cnicos e Populares.xml"
with open(filename,'r', encoding= 'utf-8') as f:
    texto = f.read()
```

Documento Glossário Termos Médicos

Etapa

- Remoção de informação prescindível manualmente , e através de expressões regulares;
- Captação das designações, posterior eliminação das mesmas do texto;
- Marcação para extração das descrições com “@@”).

```
texto = re.sub(r"</?page.*>", "", texto)  # remoção dc
texto = re.sub(r"</?text.*?>", "", texto)  # remoção d
texto = re.sub(r"</?fontspec.*?>", "", texto) # remoçã
texto = re.sub (r"<i>", "",texto) # remoção da tag itá
texto = re.sub (r"</i>", "",texto) # remoção da parte
texto = re.sub(r"<b>[A-Z]</b>", "", texto) # remoção d

# -------------- criação de lista com todas as designa

lista_designacoes = re.findall(r"<b>(.*)</b>", texto)


texto2 = re.sub(r"<b>(.*)</b>", "", texto)  # remoção

texto2 = re.sub(r"\(pop\)", "@@", texto2)    # alteraç
texto2 = re.sub(r"@@  @@", "@@" , texto2)

texto2= re.sub(r"\n+", "", texto2)
texto2 = re.sub(r"\s,", "", texto2)          # resoluçã
```

Documento Glossário Termos Médicos

Etapa

- Captação para uma lista das descrições dos termos;

- Captação da primeira descrição, visto que não é antecedida pelo marcador e, consequentemente, a regex criada não engloba. Concatenação deste elemento com a primeira lista;

- Criação dos termos e armazenamento no dicionário. Por fim, o mesmo é guardado num ficheiro json.

```
lista2 = re.findall(r"@(?:\s|,|X)(.*?)@", texto2)
lista3= re.findall(r"(^.*?)\s@", texto2) # devido à regex em cima utili
lista_descricoes = lista3 + lista2



termos = [[x, y] for x, y in zip(lista_designacoes, lista_descricoes)]

dicionario={}
dicionario = dict(termos)



out = open ("termos_glossario.json", "w", encoding="utf-8")
json.dump(dicionario, out, ensure_ascii=False, indent=4)
out.close()
```

Documento Glossário Termos Médicos

Json de termos obtido


```
{
    "micrograma": " a milionésima parte de um grama ",
    "perioral": " à volta da boca ",
    "periorbital": " à volta da órbita ",
    "perivascular": " à volta dos vasos sanguíneos ",
    "depressão": "  abaixamento, abatimento, prostração ",
    "abcesso": "abcesso, tumor ",
    "empiema": " abcesso; acumulação de pus ",
    "abdómen": " barriga, ventre ",
    "abdominal": "ventral ",
    "aberrante": "anormal ",
    "perfuração": " abertura; orifício ",
    "extracção": " ablação ",
    "castração": "  ablação dos órgãos sexuais, capação, eviração, emasculação ",
    "anastomose": " comunicação natural ou artificial entre dois vasos ou nervos ",
```

Documento WIPO Pearl Covid-19 Glossary

Etapa

- Imports e leitura do ficheiro;
- Remoção de informação não relevante e que dificultasse a extração de informação;
- Criação de marcas (início e término do nome do termo com "*" e nome da categoria a começar com "@").

```
#Remocao de linhas não necessárias
ficheiro = re.sub(r"</?page.*?>",r"",ficheiro) #serve para remover a tag page

padrao = r'<text.*font="(1|15|22)".*>.*</text>\n?'
padrao2 = r'<text.*top="(65|1131|1158|1185)".*>.*</text>\n?' #Remove texto dos cabeçalhos e rodape (nº paginas ...)
ficheiro = re.sub(padrao, r"", ficheiro)
ficheiro = re.sub(padrao2, r"", ficheiro)
ficheiro = re.sub(r'<text.*font="8".*><b>(.*)</b></text>\n?',r"*\1 *\n",ficheiro) #poe o nome do termo entre *
ficheiro = re.sub(r'<text.*font="11".*>(.*)</text>\n?', r"@\1\n", ficheiro) #poe a categoria a começar com @
ficheiro = re.sub (r"</?text.*?>", "", ficheiro)
ficheiro = re.sub (r"- ", "", ficheiro) #usado para tirar quando é quebra de linha
ficheiro = re.sub('^\s*$', r"", ficheiro, flags=re.MULTILINE) #Remove linhas vazias
ficheiro = re.sub (r"<i>", "", ficheiro)
ficheiro = re.sub (r"</i>", "", ficheiro)
```

Documento WIPO Pearl Covid-19 Glossary

Etapa

- Junção de nomes de termos que estão separados em duas linhas;
- Criação de grupos de captura;

```
# Ajuste para unir linhas de fonte "8" (nome do termo) consecutivas
ficheiro = re.sub(r'\*\n\*(.*?)\*\n', r'\1*\n', ficheiro)

padrao3 = r'\*(.*?)\s+\*\n(.*?)\n@(.*?)\n((?:(?!\*|@).)*)'
correspondencias = re.findall(padrao3, ficheiro, re.DOTALL)
```

Documento WIPO Pearl Covid-19 Glossary

```
glossario = {}
for correspondencia in correspondencias:
    termo = correspondencia[0].strip()
    descricao = correspondencia[1].strip()
    categoria = correspondencia[2].strip()
    traducoes_raw = correspondencia[3].strip()
    traducoes = re.split(r'<b>\s*(.*?)\s*</b>', traducoes_raw)[1:] #divide quando encontra uma nova lingua

    #limpeza antes de meter no documento
    termo = re.sub (r"\*", "", termo)
    termo = re.sub (r"\n", " ", termo)
    descricao = re.sub (r"<b>", "", descricao)
    descricao = re.sub (r"</b>", "", descricao)
    descricao = re.sub (r"\n", "", descricao)
    traducoes = [re.sub(r"\n", "", traducao) for traducao in traducoes]

    glossario[termo] = {
        "Descricao": descricao,
        "Categoria": categoria,
        "Traducoes": {traducoes[i].strip(): traducoes[i+1].strip() for i in range(0, len(traducoes), 2)}
    }

file_out = open("wipo.json","w",encoding= 'utf-8')
json.dump(glossario,file_out,indent=4,ensure_ascii=False)
file_out.close()
```

Documento WIPO Pearl Covid-19 Glossary

Json de termos obtido


```
{
    "acute respiratory distress syndrome": {
        "Descricao": "(syn.) ARDS   Respiratory disease characterized by the rapid onset of widespread inflammation in the lungs.",
        "Categoria": "MEDI, Pathology",
        "Traducoes": {
            "AR": "ةمزلاتمةقئاضلاةيسفنتلاة داحلا",
            "DE": "akutes Atemnotsyndrom des Erwachsenen, (syn.) ARDS",
            "ES": "síndrome de dificultad respiratoria aguda, (syn.) SDRA",
            "FR": "syndrome de détresse respiratoire aiguë, (syn.) SDRA",
            "JA": "急性呼吸窮迫症候群, (syn.) 急性呼吸促迫症候群, ARDS",
            "KO": "급성  호흡곤란  증후군, (syn.) ARDS",
            "PT": "síndrome do desconforto respiratório agudo, (syn.) SDRA",
            "RU": "острый респираторный дистресс-синдром, (syn.) ОРДС",
            "ZH": "急性呼吸窘迫综合征, (syn.) ARDS"
        }
    },
```

Documento Anatomia Geral

Etapa

- Imports e leitura do ficheiro;

```
import re
import json

filename = "anatomia geral.xml"
with open(filename,'r', encoding= 'utf-8') as f:
    texto = f.read()
```

Documento Anatomia Geral

Etapa

- Remoção de informação prescindível manualmente , e através de expressões regulares;
- Exemplos: pages, pdf2xml, fontspec e image; títulos, sub-títulos, dígitos presentes nas imagens, rodapés, notas, entre outras;

```
texto = re.sub(r"</?page.*>", "", texto)  # remoção dos pages
texto = re.sub(r"</pdf2xml>","", texto)
texto = re.sub(r"</?fontspec.*?>", "", texto)  # remoção da linha com a informação fontspec
texto = re.sub(r'<image[^>]*>', '', texto) # remoção das imagens
texto = re.sub(r"<text[^>]*\s*>\s*[A-Z]((, [A-Z])+)?<\/text>", "", texto) # remoção de linhas
texto = re.sub(r"<text[^>]*>\s*(\d+)\s*</text>", "", texto) # remoção de linhas com digitos
texto = re.sub(r'<text.*font="(1|4|8|9|10|11|12|14|15|16|17|18|19)".*>.*</text>\n',"",texto)
texto = re.sub(r'\n{2,}', '\n', texto)
texto = re.sub(r'^\s+|\s+$', '', texto, flags=re.MULTILINE)
texto= re.sub(r'<text[^>]*\sfont="3"[^>]*>\s*(.*?)\s*</text>', r'\1', texto)
```

Documento Anatomia Geral

Etapa

- marcação das designações dos termos presentes no ficheiro, utilizando a marca "@@".

```
texto = re.sub(r'<text[^>]*\sfont="([567])"[^>]*>(.*?)</text>', r'@@\2', texto)
texto = re.sub(r'(<text[^>]*\sfont="13"[^>]*>)(.*?)(</text>)', r'@@\2', texto)
texto = re.sub(r'@@<([bi])>(.*?)</\1>', r'@@\2', texto) # remoção das tags bold e italic após a marcação
```

Documento Anatomia Geral

Etapa

- utilização de uma regex para captação da informação, utilizando os macadores colocados previamente;
- atribuição das respetivas informações às variáveis designação e descrição e posterior armazenamento no dicionário;
- Por fim, o mesmo é guardado num ficheiro json.

```
# Capturar designações e descrições
matches = re.findall(r'@@\s*(.+?)(?=\n@@|$)', texto, re.DOTALL)

# Criar dicionário com designações e descrições
dicionario = {}
for match in matches:
    linhas = match.strip().split('\n')
    designacao = linhas[0]
    descricao = '\n'.join(linhas[1:]) if len(linhas) > 1 else "Sem Descrição"
    descricao = descricao.replace('\n', '')
    descricao = re.sub(r"\s[A-Z]((, [A-Z])+)?$", "",descricao)
    dicionario[designacao] = descricao


out = open ("termos_anatomia.json", "w", encoding="utf-8")
json.dump(dicionario, out, ensure_ascii=False, indent=4)
out.close()
```

Documento Anatomia Geral

Json de termos obtido


```
"Pescoço.": "Seu limite superior passa por umalinha ao longo da margem inferior da mandí-bula, processo mastóide, linha nucal superior,a
"Tronco.": "Sem Descrição",
"Tórax.": "Parte do tronco, entre o pescoço e o abdo-me. Sua estrutura básica é a caixa torácica. Seulimite inferior é a abertura toráci
"Peito.": "Sem Descrição",
"Abdome.": "Parte do tronco entre o tórax, a mar-gem superior do sacro, o ligamento inguinal e asínfi se púbica.",
"Pelve.": "Parte do tronco entre o abdome e o soalhoda pelve. A pelve maior e a pelve menor são se-paradas pela linha terminal.",
"Dorso.": "Parte posterior do tronco.",
"Membro superior.": "Constituído pelo cíngulo domembro superior e pela extremidade livre.",
"Cíngulo do membro superior.": "Sua estrutura ósseabásica é formada pela escápula e pela clavícula.",
"Axila.": "Cavidade axilar. Espaço de união entre omembro superior e a parede lateral do tórax.",
```

Junção

dos json.

Etapa

- Leitura dos 3 ficheiros e atribuição de um valor à variável glossário, que passa a ter o valor do ficheiro json “conceitos.json”;
- Verificação das keys, presentes no ficheiro e se houverem repetidas serão agrupadas senão é criada uma nova.

```
glossario = conceitos

for termo, informacao in termos.items():
    if termo in conceitos:
        # Se o termo já existir em conceitos.json, adicione uma nova chave "Descricao 2" com a nova informação
        glossario[termo]["Descricao 2"] = informacao
    else:
        # Se o termo não existir em conceitos.json, adicione-o ao dicionário conceitos com todas as suas informações
        glossario[termo] = {"Categoria": "Sem Categoria","Descricao": informacao,}

for termo, informacao in anatomia.items():
    if termo in conceitos:
        # Se o termo já existir em conceitos.json, adicione uma nova chave "Descricao Anatomia" com a nova informação
        glossario[termo]["Descricao Anatomica"] = informacao
    else:
        # Se o termo não existir em conceitos.json, adicione-o ao dicionário conceitos com todas as suas informações
        glossario[termo] = {"Categoria": "Sem Categoria","Descricao": informacao,}

file_out = open("glossario_geral.json","w",encoding= 'utf-8')
json.dump(glossario,file_out,indent=4,ensure_ascii=False)
file_out.close()
```

Json glossário geral obtido obtido


```
    "Recesso esfenoetmoidal.": {
        "Categoria": "Sem Categoria",
        "Descricao": "Espaço, em forma defenda, acima da concha nasal superior."
    },
    "Meato nasofaríngeo.": {
        "Categoria": "Sem Categoria",
        "Descricao": "Parte da cavidade nasal quese estende da mar gem posterior das conchas na-sais até os cóanos."
    },
    "Cóano.": {
        "Categoria": "Sem Categoria",
        "Descricao": "Abertura nasal posterior. As duas aber-turas entre a cavidade nasal e a parte nasal dafaringe."
    },
    "Forame esfenopalatino.": {
        "Categoria": "Sem Categoria",
        "Descricao": "Orifício superior na fossapterigopalatina que conduz à cavidade nasal. Amaior parte é formada pelo palatino, e a menor,pelo esfenóide."
    }
}
```

Conclusão.