Universidade do Minho
Escola de Engenharia

Engenharia Biomédica
Universidade do Minho

Processamento de Linguagem Natural

MEBIOM – Informática Médica 2022/2023

Dicionário Médico - melhoria, aumento e Interface

Relatório do Trabalho Prático 2

Grupo:

Ana Sá (PG49857)
Mariana Afonso Rodrigues (PG51211)
Mariana Fernandes (A92048)

Índice

Índice de Figuras

Introdução

O presente relatório serve de apoio ao trabalho desenvolvido para o trabalho prático 2 da unidade curricular de Processamento de Linguagem Natural.

Neste trabalho, pretende-se enriquecer o dicionário obtido no trabalho prático 1 pela adição de mais informações de fontes externas, como sites online (*Web scraping*) e outros dicionários médicos. O *web scraping* denomina o processo de recolha de dados estruturados da web de uma forma automatizada.

Assim, em primeiro lugar, procedeu-se à melhoria de um dos dicionários do trabalho 1 e a reorganização do dicionário final. Seguiu-se a procura de dicionários online e posteriormente ao processo de *web scraping*. Foram adicionados 3 novos dicionários, sendo que 2 destes estão relacionados com o caso de estudo escolhido para demonstrar a aplicação, que tem como tema a saúde feminina. Posteriormente, fez-se a junção dos novos dicionários com o dicionário final do trabalho prático 1 e procedeu-se a uma análise dos termos e das suas possíveis relações, de forma a agrupá-los em categorias. Por fim, foi criada uma interface que permite consultar todos os termos do dicionário e as categorias a ele associadas.

Os ficheiros utilizados foram: Dicionário de Termos Médicos português-inglês-espanhol, Glossário de Termos Médicos Técnicos e Populares e *WIPO Pearl Covid-19 Glossary*. Os links utilizados foram: Cruz Verde, *American College of Obstetricians and Gynecologists (ACOG)* e *Female Health Glossary*.

Desenvolvimento

Melhoria do trabalho 1

Como primeira melhoria, foram feitas alterações ao glossário de termos relacionados com a Covid-19, já trabalhado no primeiro trabalho prático. Estas alterações passam por aumentar o número de traduções para todas as existentes no glossário e captar, também, os sinónimos de cada termo. Além disso, também se mantiveram as *tags*, que indicam a área científica em que o termo se enquadra.

Outra diferença relativa ao primeiro trabalho, refere-se à estruturação do dicionário. A estratégia usada consiste em unir todos os dicionários, tendo em conta as chaves repetidas, de modo a organizar melhor a informação presentes nos valores. Neste âmbito, também se decidiu incluir subcategorias: Descrições e Traduções, de modo a facilitar a compreensão do mesmo, com as respetivas fontes de informação. Com isto, notou-se um aumento significativo de 773 para 5418 termos.

Aumento de informação através de *Web Scraper*

Para aumentar o conteúdo do dicionário, procedeu-se à junção do conteúdo de 3 *links*. Como será apresentado mais à frente, uma parte do trabalho foi focada em termos médicos da saúde da mulher, dando uma utilização mais específica à nossa aplicação, além da utilização geral de todos os termos obtidos e colecionados.

Relativamente aos *links*, o primeiro corresponde a um glossário de saúde da Cruz Verde [1], o segundo corresponde a um dicionário desenvolvido pelo Colégio Americano de Obstetras e Ginecologistas (ACOG) [2] e o terceiro, a glossário de saúde feminina [3].

Uma secção de código relativa à obtenção de informação da fonte *American College of Obstetricians and Gynecologists (ACOG)* é apresentada na figura abaixo.

```
url = "https://www.acog.org/womens-health/dictionary"
html = requests.get(url).text

soup = BeautifulSoup(html, "html.parser")

divs = soup.find_all("div", class_="wysiwyg-content section-terms")

dic = {}
final_dic = {}

for div in divs:

    ps = div.find_all("p")

    for p in ps:
        termo = p.find("span", class_="section-term")
        if termo:
            t = termo.text
            descricao = p.find("span", class_="section-term-definition")
            if descricao:
                desc = descricao.text

                dic[t] = desc

                translator = GoogleTranslator(source='en', target='pt')
                translated_title = translator.translate(t)
                translated_description = translator.translate(desc)

                final_dic[translated_title] = {
                    "desc_pt": translated_description,
                    "en": t,
                    "desc_en": desc
                }
```

Figura 1 - Web Scrapper com utilização de requests e BeautifulSoup

De notar que depois da obtenção da informação, foi feita ainda a tradução, pelo *deep_translator (Google Translator)* dos elementos necessários para igual formatação do dicionário final.

Junção final dos conteúdos

Para a obtenção do dicionário completo que engloba toda a informação recolhida e mencionada anteriormente, foram seguidos alguns passos:

- Junção dos dicionários relativos à Saúde da Mulher num JSON, priorizando o dicionário do ACOG aquando da junção, ou seja, mantendo o seu termo se fosse comum ao outro e adicionando o outro caso ainda não existisse no primeiro. Esta prioridade foi definida tendo em conta o número de termos ser superior no dicionário do ACOG.

- Junção do dicionário melhorado do trabalho prático 1 com o dicionário da Cruz Verde, priorizando os termos da Cruz Verde, porque o dicionário já estava todo escrito em português e, assim, priorizou-se a escrita portuguesa original em detrimento da obtida pelo *GoogleTranslate.*

- Junção do dicionário resultante do ponto acima com o resultante do primeiro ponto, priorizando os termos da Cruz Verde, obtendo-se o dicionário final a ser trabalhado na interface.

- Nas junções, foi sempre tido em conta o aumento da informação, mesmo quando os elementos já eram comuns.

Parte do código representativo da junção final dos conteúdos é apresentado na figura seguinte:

```
# 1º fase - junção do dicionário final TP1 com dicionário Cruz Verde
for key, value in dic_new.items():
    if value.get('Descrições') is None and unidecode(key.lower()) in cv_new.keys():
        value['Descrições'] = {'desc_pt':"Cruz Verde - " + cv_new[key]}
    elif value.get('Descrições') is not None and unidecode(key.lower()) in cv_new.keys():
        value['Descrições']['desc_pt'] = "Cruz Verde - " + cv_new[key]

# 2º fase - junção com o dicionário da mulher
for key, value in dic_mulher.items():
    if unidecode(key) not in dic_new.keys():
        dic[key] = value

for key, value in dic.items():
    if value.get('Descrições') is None and unidecode(key.lower()) in dic_mulher_new.keys():
        value['Descrições'] = {'desc_pt':dic_mulher_new[unidecode(key.lower())]['Descrições']['desc_pt'], 'desc_en':dic_mulher_ne
    elif value.get('Descrições') is not None and value['Descrições'].get('desc_pt') is None and unidecode(key.lower()) in dic_mul
        value['Descrições']['desc_pt'] = dic_mulher_new[unidecode(key.lower())]['Descrições']['desc_pt']
        value['Descrições']['desc_en'] = dic_mulher_new[unidecode(key.lower())]['Descrições']['desc_en']

    if value.get('Traduções') is None and unidecode(key.lower()) in dic_mulher_new.keys():
        value['Traduções'] = {'en':dic_mulher_new[unidecode(key.lower())]['Traduções']['en']}

dic = dict(sorted(dic.items(), key=lambda x: locale.strxfrm(x[0])))
```

Figura 2 - Excerto de código da junção final

Com a junção de todos os conteúdos, conseguiu totalizar-se **6127** termos.

Caso de estudo – Saúde da Mulher

Com a especificação de uma parte da aplicação num caso de estudo, nomeadamente a saúde da mulher, esta poderia assim ser uma forma mais interessante de abordar e demonstrar

diferentes aspetos relacionados à saúde feminina. Assim como, permitir o empoderamento e tomada de decisão mais informada por parte das mulheres.

Além do dicionário final resultante das várias junções de conteúdos, utilizou-se então, na interface, o dicionário resultante apenas da junção do conteúdo da Saúde da Mulher para a especificação da aplicação num dado ramo/categoria, como se pode ver na secção de Resultados.

Categorização dos termos

Para o processo de categorização dos termos presentes no dicionário, foi utilizada a biblioteca *Python Gensim*, conhecida pela sua importância na área de processamento de linguagem natural.

Com esta biblioteca, foi possível utilizar um modelo pré-treinado *Word2Vec* que, posteriormente, foi utilizado para calcular a similaridade entre as chaves do dicionário e as categorias definidas pelo grupo, relacionadas com a área da saúde. Com a similaridade calculada para cada par termo-categoria, foi possível associar cada termo à categoria que possuir maior similaridade. Como existiam termos que não se encontravam presentes no modelo escolhido, criou-se uma categoria adicional que incluísse esses termos.

Interface

Um dos objetivos deste trabalho foi a criação de uma ferramenta capaz de explorar as relações dos dados, representar as informações adequadamente e ainda permitir atualizar ou adicionar novas informações. Para cumprir esse objetivo e também para tornar a experiência da aplicação mais intuitiva e mais agradável, foi criada uma interface gráfica web com diversas páginas interligadas. Esta foi construída utilizando a *framework web Flask* do *Python, CSS, JavaScript* e *Bootstrap*.

1. Página principal

Figura 3 - Interface: página inicial

Opções disponibilizadas ao utilizador:

- Consulta de termos médicos por:
  - Dicionário (lista de termos com aba lateral para secções de letras e opção de eliminar termo)
  - Categorias
  - Tabela

- Consulta do termo e respetivas informações (Traduções e Descrições)

- Pesquisa, quer por termo (ou qualquer valor associado), quer por categoria

- Case-study (Saúde da Mulher), com barra de pesquisa embutida e opção de eliminar termo

- Adição de termo com apenas designação e descrição em português, sendo realizada automaticamente a tradução em todas as línguas já utilizadas e inserção do novo termo no dicionário principal

2. Listagem de termos como dicionário

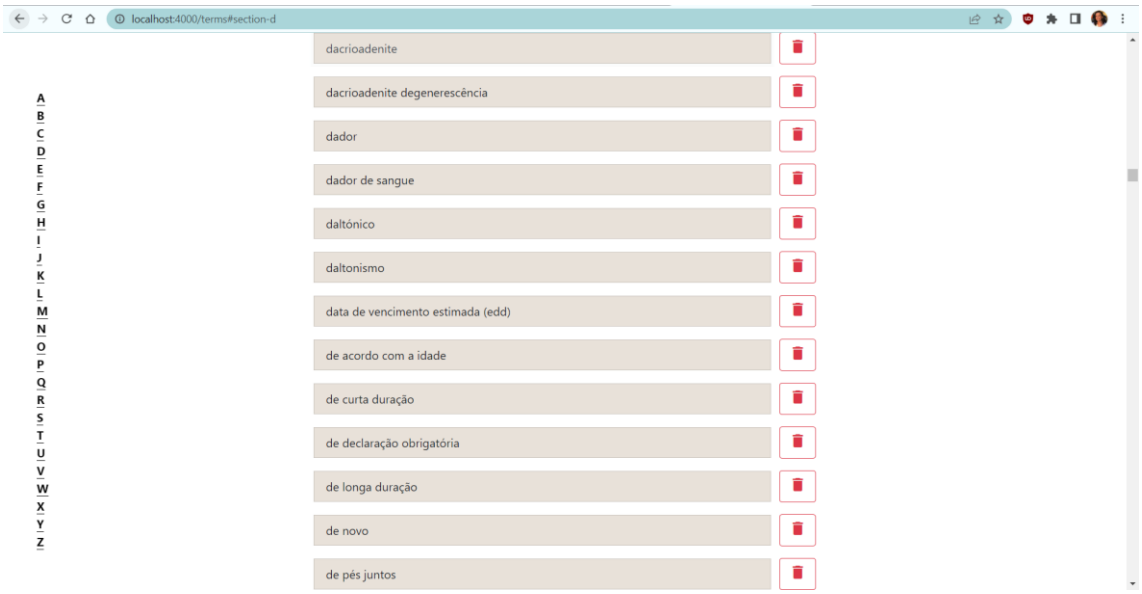

Figura 4 - Interface: listagem de termos como dicionário

3. Listagem de termos por categoria

Figura 5 - Interface: categorias de termos

4. **Tabela resumo dos termos** (com traduções principais EN, ES, FR – de modo a não sobrecarregar a visualização da mesma no caso de haver várias traduções)

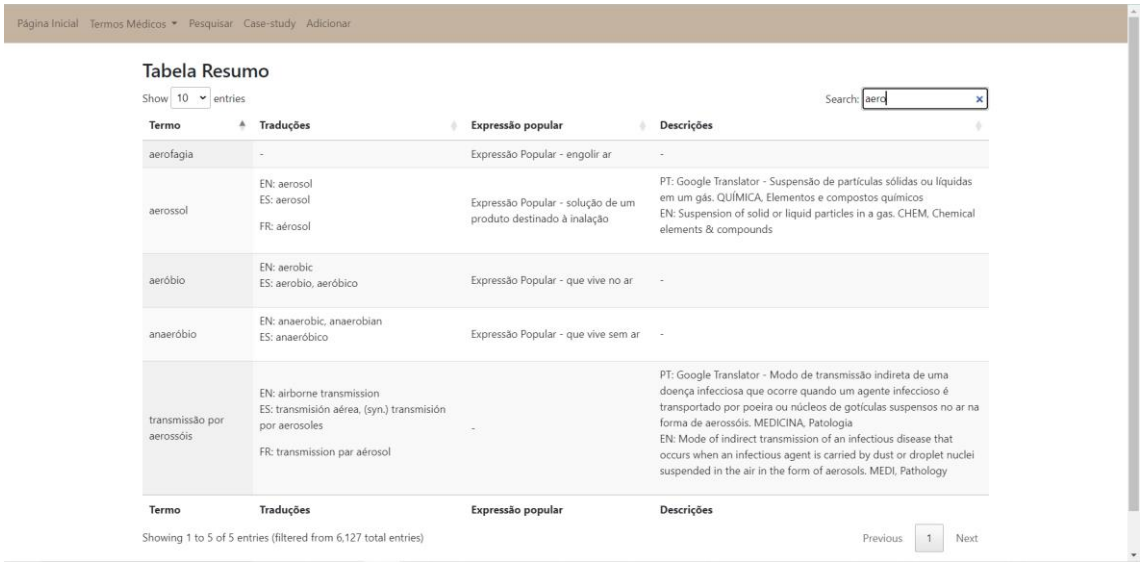

Figura 6 - Interface: tabela resumo

5. **Informação por termo**

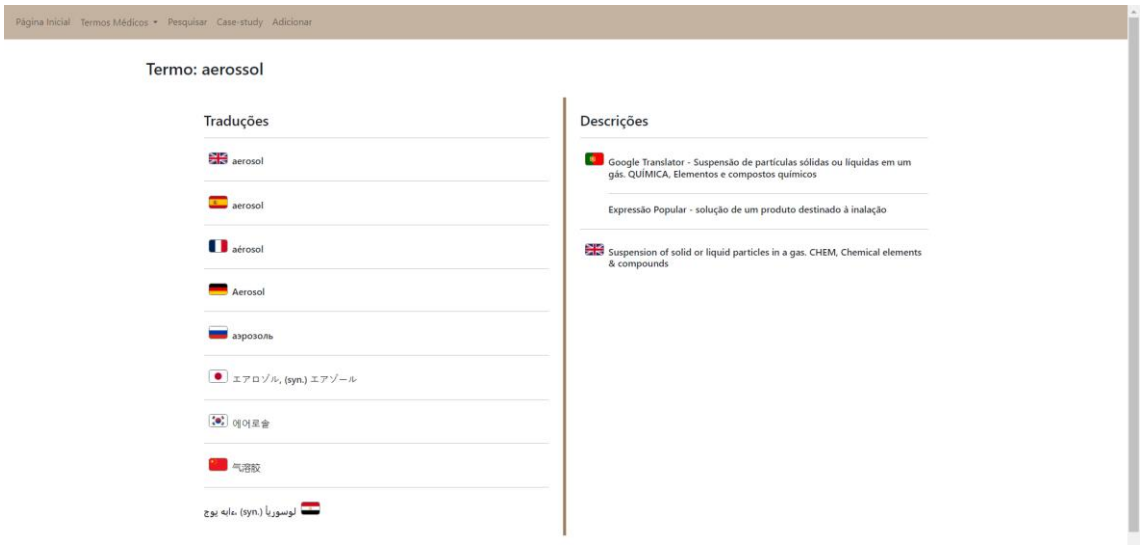

Figura 7 - Interface: informação do termo

6. Pesquisa (com validações)

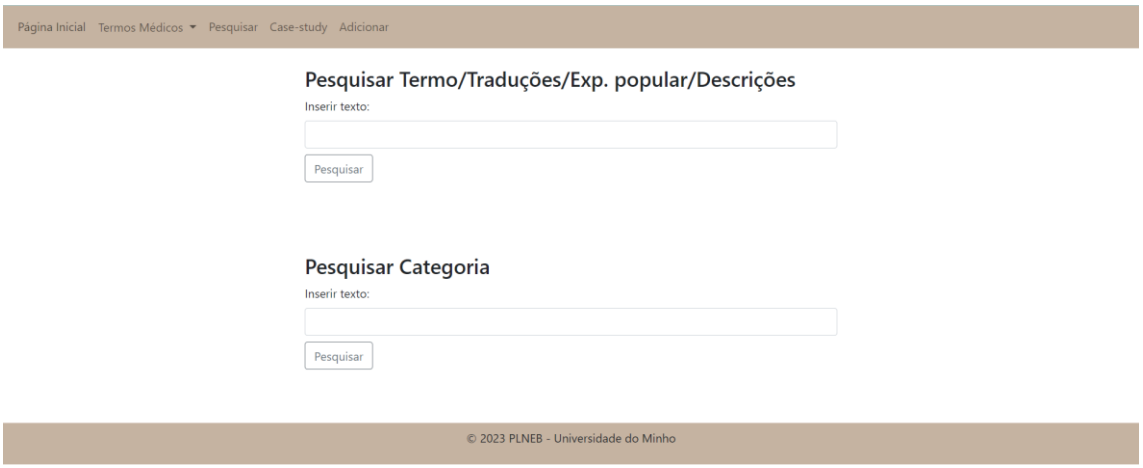

Figura 8 - Interface: página principal de pesquisa

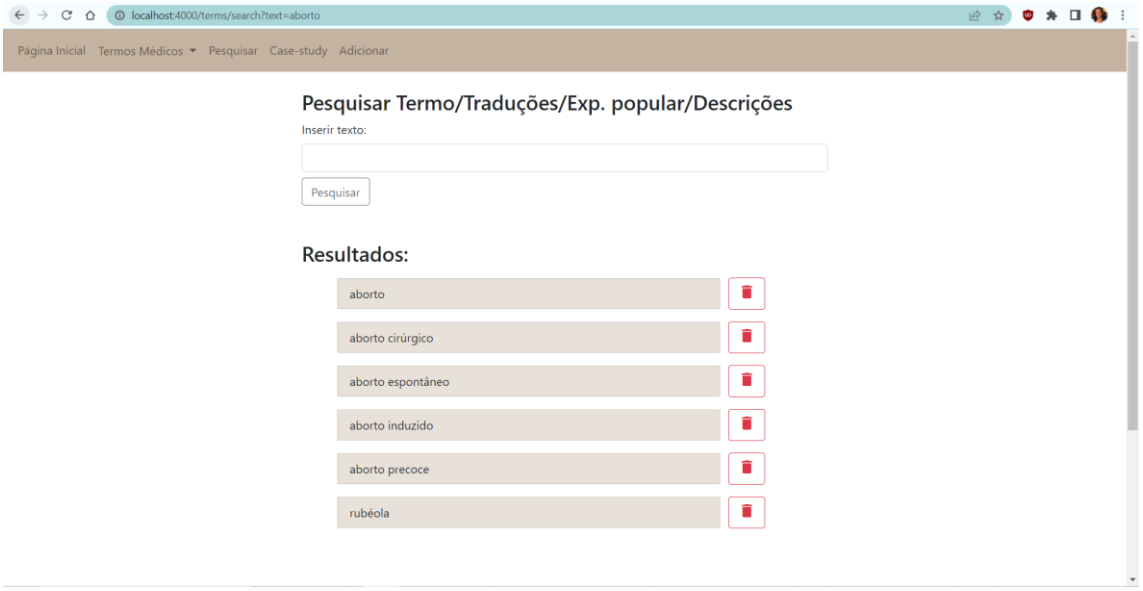

Figura 9 - Interface: resultados de pesquisa por termo/traduções/exp. pop/descrições

7. Case-study (saúde da mulher) com barra de pesquisa embutida e termos apresentados como dicionário

Figura 10 - Interface: Case-study saúde da mulher

8. Adicionar termo + Resultado

Figura 11 - Interface: adicionar termo

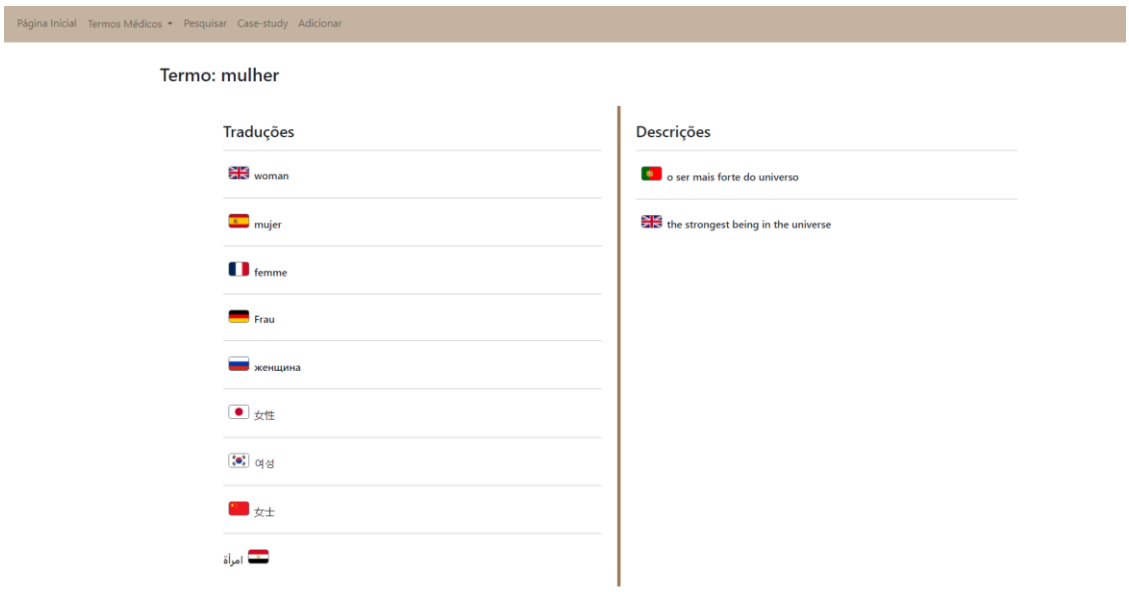

Figura 12 - Interface: resultado da adição de termo

Melhorias/ Conclusão

No futuro, como possíveis melhorias deveriam ser implementados os seguintes pontos:

- Explorar mais categorias para evitar termos sem categoria
- Explorar mais *URLs* para mais conteúdo
- Normalizar os resultados (processamento dos tokens)
- Arranjar dados suficientes para ter atributos o mais parecidos possível em todos os termos

Em suma, o desenvolvimento deste trabalho permitiu consolidar os conhecimentos relativamente ao processamento de linguagem natural através de expressões regulares e ao funcionamento do processo de *web scraping*, assim como a maior familiarização com desenvolvimento web por meio do Flask.

Referências

[1] "Cruz Verde - Serviços de Assistência Médica - Glossário de Saúde." http://www.cruzverde.pt/apoio-cliente/glossario-saude (accessed Jun. 01, 2023).

[2] "Dictionary | ACOG." https://www.acog.org/womens-health/dictionary#urinary-incontinence (accessed Jun. 01, 2023).

[3] "Female Health Glossary | Elara Care." https://elara.care/culture/female-health-glossary/ (accessed Jun. 01, 2023).