Processamento de Linguagem Natural

Flávio Ribeiro pg52290 Miguel Vale pg54094 Tayan Peller pg54451

2023-2024

1 Introduction

Em diversas áreas, especialmente em engenharia biomédica, uma gestão eficiente de informação é crucial para impulsionar a inovação, avançar na pesquisa e desenvolvimento de soluções. Neste projeto, o principal objetivo foi abordar este desafio onde se procurou extrair informações úteis de uma variedade de ficheiros em formato PDF, com estruturas e complexidade variadas. A ampla quantidade de dados disponível neste tipo de ficheiros oferece uma riqueza de informação, contudo estas quantidades de dados pode se revelar um desafio para a extração de informação considerada útil. Com isto, o nosso foco procura desenvolver uma abordagem eficaz para analisar esses documentos e extrair apenas a informação que nos é relevante, eliminando o ruído e a redundância.

2 Tratamento de documentos

De forma a atingir o objetivo proposto, a extração de informação útil, procuramos ter uma abordagem metódica que envolveu diversas etapas. Numa etapa inicial, é feito um estudo do documento, ainda em formato pdf, com intenção de retirar a estrutura e incoerências do mesmo. Já com os documentos em formato XML, passamos a uma manipulação do ficheiro e remoção de dados considerados irrelevantes, caso necessário. Num passo seguinte, passamos ao desenvolvimento em python utilizando expressões regulares (regex) para retirar concreta e exclusivamente a informação necessária de cada ficheiro. Por fim, os dados extraídos foram armazenados numa estrutura útil em JSON.

3 Glossário Termos Técnicos

A estrutura deste glossário revelou-se relativamente simples de entender estando bem explicita desde o início. Este está dividido por secções, organizadas em ordem alfabética. Em cada secção existe um respetivo número de termos e as respetivas definições dos mesmos. A ordem alfabéticanão se limita aos termos como também às definições, que aparecem antes dos termos de modo intercalado com os casos normais e é de notar que a maioria dos termos estão duplicados.

O primeiro passo feito, foi transformar este pdf num ficheiro XML. Desta forma facilita-nos a manipulação/edição do mesmo.

3.1 Tratamento do ficheiro

Sendo este um ficheiro mais simples onde não estamos deparados por fundos, imagens, legendas, número de páginas ou erros no reconhecimento de texto optamos por retirar manualmente a capa do ficheiro XML, sendo que não seria útil para a finalidade deste projeto.

O ficheiro XML é composto por um conjunto de tags de texto dentro de tags das páginas. As tags de texto tanto podem conter outras tags como bold para os termos e itálico para as definições, como também espaços vazios, vírgulas intercaladas por espaços, marcadores do final da definição denominado por (pop). O agoritmo em si é constituído por uma série de exclusões de partes do XML de modo a manter apenas a informação útil e mais complexa de remover. Em função deste objetivo foram removidas as letras referentes à ordem alfabética e os espaços vazios dentro das tags de texto. Para facilitar a utilização do marcador (pop), foram substituídas todas as suas instâncias por apenas (pop), removendo espaços e vírgulas à volta deste. Finalmente foram removidas todas as tags de texto, de página e de fontspec.

Um dos problemas encontrado foram 8 termos que não tinham uma definição associada, respetivamente: ambiente, antidopaminérgico, antifibrinolítico, bacteróide, betablocante, camada superficial, clónico, pós-carga. Como havia termos sem definições verificamos se o mesmo acontecia inversamente. Ou seja, se havia definições sem termos, o que não se verificou. Estes termos foram removidos, visto que, não ofereciam nenhuma informação útil.
Um dos maiores desafios deste ficheiro não foi propriamente a estrutura mas a falta de consistência da mesma. Mais concretamente, havia termos e definições com a ordem trocada. Foi verificado que esta consistência deveu-se à duplicação dos termos e definições, ou seja para cada (definição : termo) havia um inverso (termo : definição). O número de termos/definições presentes antes deste tratamento do ficheiro são 3639 e depois de serem removidos os duplicados sobraram 1819.
Outro grande problema no XML foi a separação de um termo em duas linhas, que no XML pode acontecer devido à má ordem de algumas definições, o que tornou quase impossível de diferenciar entre casos em que dois termos estavam seguidos por serem o mesmo em duas linhas e o caso em que um termo é apresentado depois da definição seguido de um termo que é apresentado antes da definição. Para resolver este problema utilizou-se o facto de a maioria dos termos estarem duplicados para verificar se a união de termos divididos por apenas um parágrafo, existiam noutro lugar do dicionário e se sim substituir os termos separados em duas linhas pelo termo completo.

Após estas alterações já foi possível selecionar todos os termos e guardá-los

numa lista única a estes. Para obter todas as definições foi necessário seguir um conjunto de passos para remover informação desnecessária.

Primeiramente foram removidos todos os termos com as suas tags de bold e depois foram removidos os parágrafos, de modo a ficar tudo numa linha só. Foi possível verificar que existiam conjuntos de espaços aleatórios que foram substituídos por um só. As tags de itálico que separavas linhas das mesmas definições foram substituídas por um espaço e as do início e do fim de cada definição foram apenas removidas, finalmente foram utilizados os marcadores (pop) para fazer split das definições.

Com as duas listas de termos e definições na ordem em que são apresentadas, foi criado um dicionário com letras para dividir os termos de modo alfabético e utilizando uma função própria para descobrir a primeira letra que aparece no termo este foram inseridos no dicionário corretamente, de modo a não adicionar os repetidos. Por último estes foram ordenados dentro de cada um dos seus grupos de letras de acordo com uma função lambda, para aumentar a ordenação do dicionário.

Assim, o dicionário final apresenta o seguinte formato:

$$\{A : [(termo_{1a} : definição_{1a}), ..], B : [(termo_{1b} : definição_{1b}), ..], .., Z[...]\}$$

4 Glossário Ministerio Saúde

Após as alterações do xml, foi decidido que informações neste seriam pertinentes, chegando-se à conclusão que tudo para além das siglas e das definições por ordem alfabética seriam excluídas manualmente do XML. Isto é, a capa, o sumário, a apresentação, a introdução e todas as páginas desde a página 113 até ao final do pdf.

4.1 Tratamento do ficheiro

O ficheiro XML é constituído por tags de páginas que incluém tags de texto com a informação toda do pdf Houve um conjunto de 5 diferentes alterações feitas tanto para as siglas como para o glossário, estas foram a remoção das tags fontspec e de imagens, a remoção de espaços vaizos e dos parágrafos deixados pelos espaços vazios e a eliminação de páginas sem conteúdo.

4.2 Siglas

Para a manipulação das siglas, as páginas em que estas se encontram foram guardadas numa nova variável que sofrerá um conjunto de alterações, estas, para além da utilização da função regex, re.findall() para guardar estas páginas foram, a remoção dos números de página, das tags de página e de texto e do título Siglas. Após estas alterações foi criado um marcador atrás das tags bold e estas foram posteriormente removidas. Foi feito o split pelo marcador para

separar todos os casos sigla - definição e finalmente estas foram separadas pelo símbolo '-' de modo a guardar as siglas e definições em listas diferentes de onde se vão utilizar estes dados para formar o dicionário com estas tais siglas e definições.

4.3 Glossário

O Glossário apresentava uma estrutura menos coerente devido a erros de formatação, portanto foram necessárias 21 linhas de funções regex para alterar completamente o XML. Estes incluíram a remoção das páginas das siglas, da informação no topo das páginas que eram desnecessárias, das tags de páginas, dos traços de continuação de linha nas definições, das tags de texto, do espaço entre diferentes linhas. Depois destes passos foram alterados erros encontrados no pdf, como a má escrita da categoria epidemiologia como bold e a má formatação das palavras categoria. A seguir foram resolvidos os casos em que palavras estavam tanto apresentadas em bold como em itálico o que não pode acontecer devido ao modo como o algoritmo foi organizado.

Foram criados marcadores para os termos, @, e para o início de uma nova definição, §, foram unidas continuações de descrições em diferentes linhas e as tags de itálico foram removidas, visto que, já não são necessárias. Para decidir que linhas pertenciam à descrição e quais pertenciam à categoria, foi utilizado um marcador que se encontra atrás de todas as linhas com um width superior a 228, #, visto que, esta é a largura máxima que as categorias podem apresentar. Para o caso da descrição ser pequena foram adicionados marcadores às linhas exatamente antes dos termos. Depois disto, foram removidas as tags bold e adicionada uma nova categoria 'None' para os casos em que estam não estão presentes. Mas como o último caso também não tem categoria e não se encontra presente nos casos apontados pelas funções, este foi alterado individualmente.

Como todas as linhas a não ser as das categorias têm um marcador, foi adicionado um antes de todas as linhas sem marcador, foram removidas todas as linhas que diziam Categoria tal como todas as tags de texto e todos os parágrafos. Finalmente, todos os casos de definição foram separados pelo marcador de definição §.

Depois destas alterações temos uma lista dividida por todos os casos, foram utilizadas funções re.findall(), para encontrar cada uma das informações divididas pelos marcadores criados anteriormente e após isto, foram removidos os marcadores a meio das frases.
Por fim, é criado um dicionário com as letras para manter a ordem alfabética, uma lista com a categoria e a descrição e estas foram incluídas num dicionário com o termo referente a cada um. Foram criados dois ciclos para iterar sobre o dicionário das letras e o dicionário das definições para adicionar cada termo à sua letra referente e para terminar este dicionário maior foi adicionado ao dicionário criado antes para as siglas, tendo assim um dicionário final com a

seguinte estrutura.

{'Siglas':{(sigla):descrição}, 'Glossário': {(letra):{(termo):[(marcador),(definição)}}}

5 Minidicionário

O dicionário fornecido é especificamente projetado para terminologia médica e consiste em termos e conceitos relacionados à medicina, saúde e à profissão médica. A formatação é estruturada para organizar e categorizar termos, incluindo suas definições, sinônimos, antônimos, termos relacionados, atributos e referências cruzadas. Isso ajuda os usuários a entenderem os significados e as relações entre termos e conceitos, facilitando o uso eficaz do dicionário.

Algumas características-chave deste dicionário incluem:

- **Organização estruturada:** O dicionário é organizado em uma estrutura hierárquica, com termos pai e filhos. Isso permite que os usuários naveguem facilmente e explorem as relações entre diferentes conceitos.
- **Definições:** Cada termo é fornecido com uma definição que descreve seu significado. Isso ajuda os usuários a entenderem os conceitos básicos e os significados dos termos.
- **Sinônimos:** O dicionário fornece uma lista de sinônimos para cada termo. Estas são palavras ou frases que têm o mesmo significado que o termo. Os usuários podem facilmente ver que esses sinônimos podem ser usados de forma intercambiável.

No geral, este dicionário serve como um recurso valioso para profissionais de saúde, estudantes e pesquisadores. Ao organizar e categorizar termos e seus conceitos relacionados, ele permite que os usuários acessem e entendam facilmente informações sobre uma ampla gama de assuntos médicos.

5.1 Tratamento do ficheiro

Inicialmente, o código abre o arquivo XML minidicionario.xml e lê seu conteúdo em uma variável de string chamada texto. Em seguida, são removidas tags e marcações desnecessárias do arquivo XML usando a função re.sub() coom Expressões Regulares específicas. Além disso, divide-se o conteúdo dos dois dicionários, com base na marcção '#', isso ajuda a separar o conteúdo do arquivo em seções mais gerenciáveis posteriormente.

Em seguida, foi necessário substituir determinadas tags pelo símbolo '@', para agrupar termos e definições no arquivo. Criam-se duas listas que conterão os termos e definições para traduções de inglês para português e de português para inglês, respectivamente. Em seguida, cria-se duas listas vazias, termo e definicao, para armazenar os termos e definições, separados individualmente.

O código então itera pelas listas en_pt e pt_en, extraindo os termos e definições e armazenando-os nas listas termo e definicao. O código cria dois dicionários, dic_en_pt e dic_pt_en, para armazenar os termos e definições em um formato de par chave-valor.

Por fim, combina-se os dicionários dic_en_pt e dic_pt_en em um único dicionário chamado dicionario. O dicionario é então salvo em um arquivo JSON usando a função json.dump()
Estrutura:

$$\{Inglês–Português : nome : descrição\}$$

$$\{Português–Inglês : nome : descrição\}$$

$$\{EN\_PT : Inglês–Português, PT\_EN : Português–Inglês\}$$