

UNIVERSIDADE DO MINHO

Trabalho Prático 1

Elaborado no âmbito da Unidade Curricular de
Processamento de Linguagem Natural em Engenharia Biomédica

Discentes

Mónica Carina Costa Martins (a95918)

Ricardo Lopes Araújo (a97311)

Docentes

José João Antunes Guimarães Dias Almeida

Luís Filipe Costa Cunha

abril de 2024

Índice

1	Introdução	3
2	Enquadramento	3
3	Metodologias, Materiais e Métodos	3
4	Implementação	3
4.1	glossario_ministerio_saude.pdf	4
4.1.1	Análise do documento	4
4.1.2	Processamento do documento	5
4.1.3	Estrutura JSON	11
4.2	Minidicionário de Cardiologista Autor Ricardo Silveira Mello.pdf	15
4.2.1	Análise do documento	15
4.2.2	Processamento do documento	15
4.2.3	Estrutura JSON	17
4.3	medicina.pdf	18
4.3.1	Análise do documento	18
4.3.2	Processamento do documento	20
4.3.3	Estrutura JSON	25
5	Conclusão	27

1 Introdução

O presente trabalho foi desenvolvido no âmbito da Unidade Curricular de Processamento de Linguagem Natural em Engenharia Biomédica e o seu objetivo consiste na aplicação dos conhecimentos adquiridos durante as aulas acerca do processamento de documentos e extração de informação dos mesmos.

Deste modo, este projeto consiste na extração de informação de documentos com vários tipos de informação médica. Os documentos foram processados, com o auxílio de expressões regulares. Posteriormente, essa informação foi armazenada em documentos de formato JSON.

2 Enquadramento

O processamento de linguagem natural (*Natural Language Processing - NLP*) é um campo da inteligência artificial que permite extrair vários tipos de informação de textos não estruturados.

As expressões regulares (também conhecidas como *regex*) têm-se mostrado fundamentais e eficazes no pré-processamento de textos e, por isso, úteis para o processamento de linguagem natural. O facto de encontrarem e extraírem padrões específicos em textos é uma mais valia para a preparação e limpeza de documentos.

3 Metodologias, Materiais e Métodos

Durante a realização deste trabalho foram utilizados diferentes materiais e métodos para atingir o objetivo pretendido.

Primeiramente, foi necessário definir os documentos que seriam utilizados durante o projeto. Para além do documento obrigatório ("glossario_ministerio_saude.pdf"), escolheram-se os documentos "medicina.pdf" e "Minidicionário de Cardiologista Autor Ricardo Silveira Mello.pdf". Uma vez que o documento obrigatório contém termos médicos e respetiva definição, achou-se pertinente explorar documentos que fossem constituídos por conceitos e respetivas traduções.

Uma vez que se trata de documentos em formato PDF, após a sua escolha, foi necessário convertê-los para documentos de texto. Assim, utilizou-se o comando *pdftohtml -xml* que permite a conversão dos documentos para um formato HTML e, posteriormente, em XML. Esta conversão foi fundamental para manter a estrutura e conteúdo do PDF e, ao mesmo tempo, facilitar a análise e extração de informações importantes presentes no documento.

A extração de informação, por sua vez, foi possível com recurso à linguagem *Python* e às bibliotecas *re* e *json*. A primeira biblioteca permite a utilização de expressões regulares para encontrar e alterar informação nos documentos XML. Por outro lado, a biblioteca *json* permite estruturar a informação extraída em formato JSON.

4 Implementação

Para atingir o objetivo deste trabalho, todos os documentos foram primeiramente analisados (tanto antes como depois da conversão), para compreender a forma como a informação estava estruturada. Após essa análise, o documento XML resultante da conversão foi processado com expressões regulares e avaliou-se, paralelamente, a qualidade do processamento e da informação extraída.

De seguida serão explicadas as decisões tomadas durante o processamento dos diferentes documentos, bem como as estratégias adotadas para identificar a informação relevante a ser extraída dos mesmos.

4.1 glossario_ministerio_saude.pdf

4.1.1 Análise do documento

Após a análise detalhada do documento, verificou-se que o mesmo apresenta 3 secções com informação bastante relevante que deveria ser extraída:

- Siglas (páginas 5 a 9, inclusive): Esta informação está estruturada da seguinte forma:

SIGLA - Significado da Sigla

- Glossário (páginas 15 a 106, inclusive): Contém vários conceitos e expressões médicas e respetiva definição. Em maior parte dos casos, apresenta também a categoria (Área temática da BVS Saúde Pública) a que o termo pertence. Quando isso não acontece, é recomendado ao leitor que consulte outro termo. Assim, há duas formas de estruturação:

Conceito

Categoria: Área temática da BVS Saúde Pública
Definição do conceito.

Conceito

Ver outro conceito.

- Áreas temáticas da BVS Saúde Pública (páginas 107 a 112, inclusive): Esta parte apresenta a descrição das diferentes categorias anteriormente utilizadas para agrupar os conceitos. Está estruturada da seguinte forma:

Área/categoria

Descrição da área temática ou categoria.

Foi possível reparar em algumas exceções às regras apresentadas anteriormente, que poderiam precisar de uma atenção especial, nomeadamente:

- Falta de dois pontos após "*Categoria*", como é o caso dos conceitos "Solvente orgânico" e "Sistemas Formais de Cuidados";
- Categoria do conceito "Notificação de doenças" a negrito;
- Conceito "Sistema de Informação sobre Vigilância Alimentar e Nutricional (Sisvan)" escrito com tamanho de letra inferior;
- Conceitos que apresentam duas ou mais categorias, separadas pelo carácter ●, por exemplo:

Categoria: Administração e Planeamento em Saúde ● Ciência e Tecnologia em Saúde

Após esta análise do documento em formato PDF, foi também necessário observar como estava estruturado o documento XML obtido após a conversão.

Observou-se rapidamente que os termos, quer eles pertencessem ao glossário ou às áreas temáticas, estavam escritos com um tamanho de fonte igual a 21. Deve notar-se que, com

referido anteriormente, o termo "Sistema de Informação sobre Vigilância Alimentar e Nutricional (Sisvan)" estava escrito com tamanho de letra inferior. Durante esta análise, verificou-se que este termo estava escrito com um tamanho de fonte igual a 13.

O caracter ☉ observado anteriormente e responsável por dividir as diferentes categorias de um mesmo conceito, foi convertido para um elemento HTML *text*, com tamanho de fonte igual a 24, sem qualquer tipo de texto escrito no mesmo.

Para além disso, foi possível analisar que elementos HTML cujo parâmetro *top* era igual a 233, 240, 250, 247, 246, 238 ou 257 referiam-se aos cabeçalhos das páginas (Figura 1).

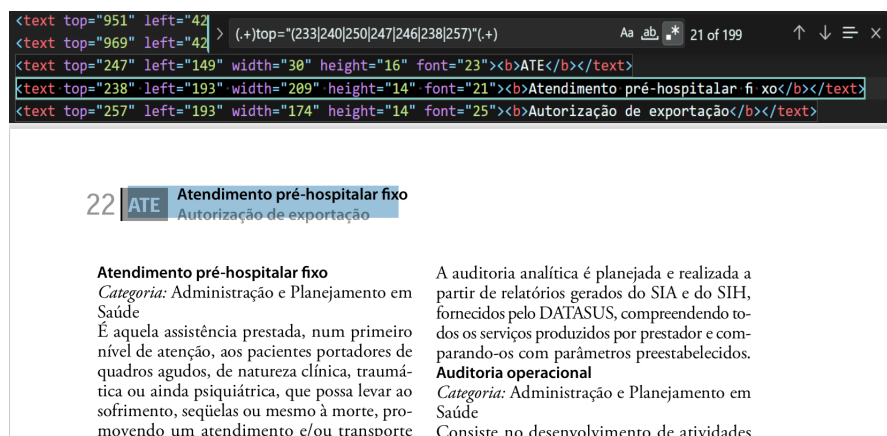


Fig. 1: Identificação do cabeçalho.

4.1.2 Processamento do documento

Como referido anteriormente, o processamento dos documentos foi efetuado com o recurso a expressões regulares.

O primeiro passo deste processamento foi a marcação das categorias, com a marca "@CAT". Para isso, utilizaram-se os seguintes padrões:

```
1 # casos em que categoria nao tem dois pontos
2 texto = re.sub(r"<.+?>Categoria<.+<n<.+>([^\n]+)<.+?>", r"\n@CAT\1\n", texto)
3
4 # correcao do caso "notificacao de doencas" que tem a categoria a bold
5 texto = re.sub(r"<.+?>Categoria: .+<n<.+?>(.+)<.+>", r"\n@CAT\1\n", texto)
```

Posto isto, o resultado obtido desta substituição e marcação já permitia identificar claramente uma separação entre os conceitos e as respetivas categorias (Figura 2).

Na Figura anterior, pode verificar-se que existiam alguns termos que ocupam duas linhas e, por isso, dois elementos *text* com tamanho de fonte igual a 21, consecutivamente (como é o caso do conceito "Acompanhamento do crescimento e desenvolvimento infantil"). Assim, estes casos foram corrigidos e marcados. Foi também corrigido o caso específico do conceito "Sisvan", referido anteriormente. Estes conceitos foram marcados com "@-".

```

<text top="363" left="427" width="237" height="14" font="21"><b>Acompanhamento do crescimento e </b></text>
<text top="381" left="427" width="162" height="14" font="21"><b>desenvolvimento infantil</b></text>

@CATAtenção à Saúde

<text top="416" left="427" width="289" height="16" font="14">Garantir a melhoria da qualidade de vida das </text>
<text top="434" left="427" width="285" height="16" font="14">crianças, permitindo pôr em evidência, pre-</text>
<text top="452" left="427" width="291" height="16" font="14">cocemente, qualquer transtorno que afete </text>
<text top="470" left="427" width="289" height="16" font="14">sua saúde e, fundamentalmente, sua nutrição </text>
<text top="488" left="427" width="157" height="16" font="14">e sua capacidade mental.</text>
<text top="507" left="427" width="110" height="14" font="21"><b>Aconselhamento</b></text>

@CATAtenção à Saúde

<text top="542" left="427" width="290" height="16" font="14">Processo de escuta ativa, individualizado e </text>
<text top="560" left="427" width="290" height="16" font="14">centrado no cliente. Pressupõe a capacidade </text>
<text top="578" left="427" width="289" height="16" font="14">de estabelecer uma relação de confiança entre </text>
<text top="596" left="427" width="285" height="16" font="14">os interlocutores, visando ao resgate dos re-</text>
<text top="614" left="427" width="289" height="16" font="14">cursos internos do cliente para que ele mesmo </text>
<text top="632" left="427" width="290" height="16" font="14">tenha possibilidade de reconhecer-se como </text>
<text top="650" left="427" width="285" height="16" font="14">sujeito de sua própria saúde e transformação.</text>
<text top="669" left="427" width="164" height="14" font="21"><b>Aconselhamento coletivo</b></text>

@CATAtenção à Saúde

```

Fig. 2: Marcação da categoria.

```

1 # marcação dos termos que ocupam duas linhas
2 texto = re.sub(r".+21\\><b>(.)\\n<. +21\\>?<b>(.)<.+\"", r"@-\\1\\2\"", texto)
3
4 # caso específico do termo sisvan escrito com letra inferior
5 texto = re.sub(r".+13\\><b>(.)\\n<. +13\\>?<b>(.)<.+\"", r"@-\\1\\2\"", texto)

```

Parte do resultado desta correção está apresentada na Figura 3, onde se pode verificar a correção do conceito "Acompanhamento do crescimento e desenvolvimento infantil").

```

@-Acompanhamento do crescimento e </b></text>desenvolvimento infantil</b>

@CATAtenção à Saúde

<text top="416" left="427" width="289" height="16" font="14">Garantir a melhoria da qualidade de vida das </text>
<text top="434" left="427" width="285" height="16" font="14">crianças, permitindo pôr em evidência, pre-</text>
<text top="452" left="427" width="291" height="16" font="14">cocemente, qualquer transtorno que afete </text>
<text top="470" left="427" width="289" height="16" font="14">sua saúde e, fundamentalmente, sua nutrição </text>
<text top="488" left="427" width="157" height="16" font="14">e sua capacidade mental.</text>
<text top="507" left="427" width="110" height="14" font="21"><b>Aconselhamento</b></text>

@CATAtenção à Saúde

<text top="542" left="427" width="290" height="16" font="14">Processo de escuta ativa, individualizado e </text>
<text top="560" left="427" width="290" height="16" font="14">centrado no cliente. Pressupõe a capacidade </text>
<text top="578" left="427" width="289" height="16" font="14">de estabelecer uma relação de confiança entre </text>
<text top="596" left="427" width="285" height="16" font="14">os interlocutores, visando ao resgate dos re-</text>
<text top="614" left="427" width="289" height="16" font="14">cursos internos do cliente para que ele mesmo </text>
<text top="632" left="427" width="290" height="16" font="14">tenha possibilidade de reconhecer-se como </text>
<text top="650" left="427" width="285" height="16" font="14">sujeito de sua própria saúde e transformação.</text>
<text top="669" left="427" width="164" height="14" font="21"><b>Aconselhamento coletivo</b></text>

@CATAtenção à Saúde

```

Fig. 3: Correção de termos que ocupam duas linhas.

Os seguintes passos do processamento consistiram na eliminação da numeração das páginas e de elementos itálicos, que iriam interferir na identificação de conceitos (como é o caso do conceito Equivalência in vitro). Para além disso, removeram-se também os conteúdos referentes ao cabeçalho das páginas (Figura 1).

```
1 # eliminacao da numeracao de paginas e indices
2 texto = re.sub(r"<.+>[0-9]{1,3}<.+>", r"\n", texto)
3
4 # remover italicos para nao interferir na sinalizacao de termos
5 # (ex: conceito "equivalencia in vitro")
6 texto = re.sub(r"<[/]?i>", r"", texto)
7
8 # remover cabecalhos das paginas com termos/conceitos
9 # (ex: ADJ Adjuvante farmaceutico Aids pediatria)
10 texto = re.sub(r"(.+)top=\"(233|240|250|247|246|238|257)\"(.+)\", r"", texto)
```

Posto isto, procedeu-se à marcação das restantes categorias e conceitos, tendo em atenção os casos específicos notados anteriormente.

```
1 # marcacao de termos que ocupam apenas uma linha
2 texto = re.sub(r"(.+)font=\"21\"><b>(.+)</b>(.+)\", r"\n@-2\n", texto)
3
4 # marcacao de categorias
5 texto = re.sub(r"<.+>Categoria:.\n<.+>(.+)\", r"\n@CAT\1\n", texto)
6
7 # casos em que os dois pontos estao separados da palavra "Categoria"
8 # (ver termo CNS)
9 texto = re.sub(r"<.+>Categoria.\n<.+>:(.+)\", r"\n@CAT\1\n", texto)
```

O resultado obtido destas modificações está apresentado na Figura 4. Pode verificar-se que a estrutura do documento apresenta melhorias significativas no que toca ao aspeto do glossário.

Nesta fase, é importante verificar que há palavras que não cabem numa linha e, por isso, estão separadas por hífen (como é o caso da palavra "melhoria", da Figura 4). É importante lidar com estes casos de maneira diferente aos restantes. Assim, removeram-se os hífen e juntaram-se as diferentes partes com a seguinte expressão regular:

```
1 # tratamento de palavras com hifens
2 texto = re.sub(r"<.+>(.+)-.+<\n", r"\1", texto)
3
4 # remover elementos html
5 texto = re.sub(r"<.+>(.+)\n", r"\1\n", texto)
```

Verifica-se, na Figura 5, que os hífen foram removidos e que as diferentes linhas que contêm as definições dos conceitos se encontram apenas à distância de um elemento *newline*.

O próximo passo consistiu na identificação de casos em que o mesmo conceito apresenta uma ou mais categorias. Nestes casos, como já foi visto anteriormente, as categorias são separadas por um elemento *text*, sem qualquer tipo de conteúdo, cujo tamanho da fonte é igual a 24. Assim, foi necessário implementar a seguinte substituição:

```
1 # termos com duas ou mais categorias
2 texto = re.sub(r"^[A-Za-z]?<\n+<.font=\"24\"></text>\n*^[A-Za-z]?(.+)\", r"\n@CAT\1\n", texto)
```

```

@-Gravidez de alto risco

<text top="326" left="176" width="167" height="16" font="14">Ver Gestaç o de alto risco.</text>

@-Grupo de apoio ao idoso

@CATAtenç o   Sa de

<text top="380" left="176" width="285" height="16" font="14">Grupo que promove a  es que visem   me-</text>
<text top="398" left="176" width="245" height="16" font="14">lhoria da qualidade de vida dos idosos.</text>

@-Grupo matricial

@CATCi ncias Sociais em Sa de

<text top="452" left="176" width="290" height="16" font="14">Grupo composto por lideran as l sbicas do </text>
<text top="470" left="176" width="285" height="16" font="14">pa s, fi liadas a ONGs que desenvolvem tra-</text>
<text top="488" left="176" width="289" height="16" font="14">balhos no  mbito da promo  o da sa de, da </text>
<text top="506" left="176" width="289" height="16" font="14">visibilidade l sbica e do combate   epidemia </text>
<text top="524" left="176" width="285" height="16" font="14">do HIV/DST. Foi criado pela CN-DST/</text>
<text top="542" left="177" width="289" height="16" font="14">AIDS em 2001, para a  es de preven  o das </text>
<text top="560" left="176" width="289" height="16" font="14">DST/aids junto  s mulheres que fazem sexo </text>
<text top="578" left="176" width="146" height="16" font="14">com mulheres (MSM).</text>

```

Fig. 4: Marca  o de categorias e conceitos.

```

@-Gravidez de alto risco

Ver Gesta  o de alto risco.

@-Grupo de apoio ao idoso

@CATAten  o   Sa de

Grupo que promove a  es que visem   melhoria da qualidade de vida dos idosos.

@-Grupo matricial

@CATCi ncias Sociais em Sa de

Grupo composto por lideran as l sbicas do
pa s, fi liadas a ONGs que desenvolvem trabalhos no  mbito da promo  o da sa de, da
visibilidade l sbica e do combate   epidemia
do HIV/DST. Foi criado pela CN-DST/
AIDS em 2001, para a  es de preven  o das
DST/aids junto  s mulheres que fazem sexo
com mulheres (MSM).

```

Fig. 5: Remo   o de h fens e elementos html.

Ap  s esta substitui   o, verificou-se a exist  ncia de letras isoladas (por exemplo, A), referentes ao cabe  alho das p  ginas onde iniciam os termos que come  am por essa letra. Um exemplo desse cabe  alho est   na Figura 6.

Posto isto, foi necess  rio eliminar estes casos para n  o interferirem nos termos ou conceitos.

<p>Abordagem médica tradicional do adulto hospitalizado</p> <p>Categoria: Atenção à Saúde</p> <p>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</p>	<p>Ação racional</p> <p>Categoria: Atenção à Saúde</p> <p>Modelo de intervenção centrado no indivíduo no qual permite a relação entre a epidemiologia e a dimensão sociocultural do trabalho de prevenção.</p> <p>Acidentes ampliados</p>
--	--

Fig. 6: Exemplo de cabeçalho.

Para além disso, foram corrigidos casos em que as categorias se encontravam separadas, por ocuparem mais do que uma linha. Na Figura 7, pode verificar-se um desses casos.

```
@CATRecursos Humanos em Saúde Pública
blica
Vinculados em geral a universidades, esses
pólos articulam uma ou mais instituições voltadas para a formação, capacitação e educação permanente dos recursos humanos para
a saúde, em conjunto com as Secretarias de
Saúde dos estados e municípios.

@-População economicamente ativa

@CATDemografi a
```

Fig. 7: Categoria "Recursos Humanos em Saúde Pública, dividido.

Posto isto, foram implementadas as seguintes substituições:

```
1 # eliminacao de cabecalhos
2 texto = re.sub(r"\n[A-Z]\n", r"\n", texto)
3
4 # correcao da categoria Recuros Humanos em Saude Pu-blica
5 texto = re.sub(r"@CAT(.+)\n+([\s\n@<]+\n", r"@CAT\1\2\n", texto)
6
7 # correcao de outras categorias separadas
8 texto = re.sub(r"@CAT(.+)\n+([\s\n@<]+\n", r"@CAT\1\2\n", texto)
9
10 # correcao de casos em que a marca @CAT
11 # nao esta pegada a categoria
12 texto = re.sub(r"@CAT(.+)\n+@CAT\s+([\s\n@<]+\n", r"@CAT\1\n@CAT\2\n", texto
    )
```

Os próximos passos consistiram na correção de conceitos, categorias ou definições, que continham as expressões "in vivo" ou "in vitro".

```
1 # remocao do termo in vitro do cabecalho da pag 83
```

```

2 texto = re.sub(r"@CAT\s\n+<b>in vitro</b>", r"", texto)
3
4 # correcao dos restantes termos que usam "in vitro" ou "in vivo"
5 texto = re.sub(r"\n+(<.+>)?\n+<b>(in vi.+)</b>", r"\2", texto)

```

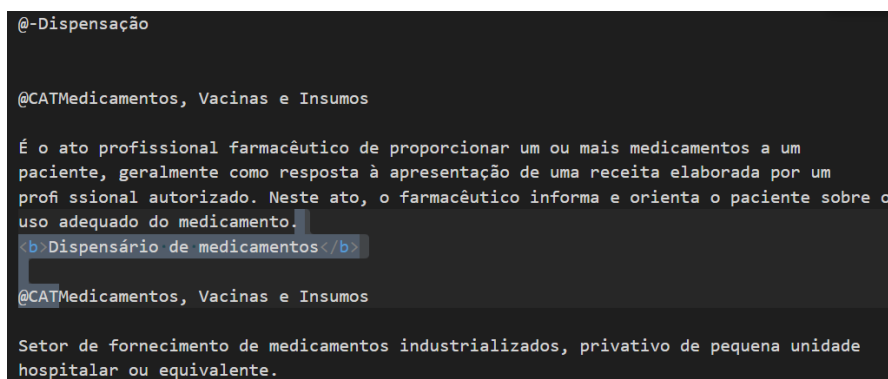
De seguida, foram realizadas algumas substituições em prol de corrigir erros existentes em algumas categorias.

```

1 # correcao de categorias como Atencao a Saude
2 texto = re.sub(r"@CAT([^\n]+)([\n]?)\n([~A-ZE0I@~\n<])", r"@CAT\1\3", texto)
3
4 # remocao de espacos entre marca e categoria
5 texto = re.sub(r"@CAT\s(.+)", r"@CAT\1", texto)
6
7 # correcao da categoria Promocao e Educacao em Saude
8 texto = re.sub(r"Promocao\s*e\s*[\n]*Ed(.+)", r"Promocao e Ed\1", texto)
9
10 # correcao da categoria Medicamentos, Vacinas e Insumos
11 texto = re.sub(r"@CAT(.+),\s*\n(.+)", r"@CAT\1, \2", texto)

```

Neste ponto, apenas resta eliminar os elementos HTML ainda existentes no documento. No entanto, não serão apagados os elementos *bold*, já que estes irão permitir a identificação das siglas. No entanto, existem conceitos que apresentam elementos *bold* e que, por isso, não estão identificados. Na Figura 8, pode verificar-se um desses casos.



```

@-Dispensação

@CATMedicamentos, Vacinas e Insumos

É o ato profissional farmacêutico de proporcionar um ou mais medicamentos a um
paciente, geralmente como resposta à apresentação de uma receita elaborada por um
profissional autorizado. Neste ato, o farmacêutico informa e orienta o paciente sobre o
uso adequado do medicamento.
Dispensário de medicamentos

@CATMedicamentos, Vacinas e Insumos

Setor de fornecimento de medicamentos industrializados, privativo de pequena unidade
hospitalar ou equivalente.

```

Fig. 8: Conceito com elemento bold.

Posto isto, fizeram-se as seguintes substituições, juntamente com outras ainda acerca da correção de categorias e conceitos:

```

1 # eliminar todos elementos HTML menos os bold
2 texto = re.sub(r"<[^b].+[^b]>", r"", texto)
3
4 # correcao e marcacao de termos com elementos a bold
5 texto = re.sub(r"\n<b>(.)</b>\n+@CAT", r"\n@-\1\n@CAT", texto)
6

```

```

7 # correcao dos termos Profae e Alimento in natura
8 texto = re.sub(r"@-(.)[\n]*@-(.)", r"@-\1\2", texto)
9
10 # correcao da categoria Vigilancia em Saude
11 texto = re.sub(r"@CAT \n+(.)", r"@CAT\1\n", texto)
12
13 # correcao de um caso especifico com duas categorias ligadas
14 texto = re.sub(r"@CAT(.)Atencao\s\sa\sSaude", r"@CAT\1\n@CATAtencao a Saude\n",
    texto)

```

Após todas estas substituições, os conceitos e respetivas definições estão prontos a ser extraídos para uma estrutura JSON, bem como as siglas. Para isso, foi utilizado o findall:

```

1 # termos com 1 categoria
2 conceitos1 = re.findall(r'@-([\~@]+)@CAT(.)[\n]*([\~@]+)', texto)
3
4 # termos com 2 categorias
5 conceitos2 = re.findall(r'@-([\~@]+)@CAT([\~@]+)@CAT(.)[\n]*([\~@]+)', texto)
6
7 # termos com 3 categorias
8 conceitos3 = re.findall(r'@-([\~@]+)@CAT([\~@]+)@CAT([\~@]+)@CAT(.)[\n]*([\~@]+)',
    , texto)
9
10 # termos sem categoria
11 conceitos4 = re.findall(r'@-(.)[\n]+Ver([\~@]+)', texto)
12
13 # siglas
14 siglas = re.findall(r"<b>(.)\s-\s?</b>\n([\~<]*)", texto)
15 siglas2 = re.findall(r"<b>(.)</b>\n[\s]?-([\~<]\*)", texto)

```

Após esta extração, as listas obtidas do código anterior foram transformadas em dicionários. Esses dicionários foram, por fim, carregados para um ficheiro JSON.

De seguida é explicada a sintaxe da estrutura de dados utilizada para representar a informação extraída do texto.

4.1.3 Estrutura JSON

No que diz respeito aos conceitos, categorias e definições retiradas da parte do glossário, optou-se pela seguinte estrutura JSON:

```

1 {
2     "Conceito1": {
3         "Categoria": ["Categoria1",
4             "Categoria2"],
5         "Descricao": "Descricao do conceito1."
6     },
7
8     "Conceito2": {
9         "Categoria": ["Categoria1"],
10        "Descricao": "Descricao do conceito2."
11    },

```

```

12
13     "Conceito3": {
14         "Descricao": "Ver conceito1."
15     }
16 }

```

Pode verificar-se que esta estrutura aborda todas as formas diferentes sob as quais os conceitos e respectivas categorias e definições estão presentes no documento (ou seja, casos em que há nenhuma, uma ou mais categoria(s)).

Por outro lado, as siglas foram extraídas para um documento JSON com a seguinte sintaxe:

```

1 {
2     "Sigla1": "Significado da Sigla1",
3     "Sigla2": "Significado da Sigla2",
4     "Sigla3": "Significado da Sigla3"
5 }

```

Nas Figuras 9 e 10 podem verificar-se excertos dos documentos JSON obtidos relativamente às siglas e aos conceitos presentes no glossário.

```

"Azitodimidina": {
    "Descrição": "Ver AZT."
},
"AZT": {
    "Categoria": [
        "Medicamentos, Vacinas e Insumos"
    ],
    "Descrição": "Sigla do composto farmacológico azitotimidina. Também conhecida",
},
"Baixo peso ao nascer": {
    "Categoria": [
        "Alimentação e Nutrição",
        "Atenção à Saúde",
        "Epidemiologia"
    ],
    "Descrição": "Classificação de recém-nascidos com menos de 2.500g."
},

```

Fig. 9: Documento JSON com os conceitos e respectivas categorias e definições.

Verificou-se ainda que, no total, foram extraídos 705 conceitos e 224 siglas.

Para extrair as definições das diferentes áreas temáticas, foi duplicado o documento XML original e removidos os capítulos "Siglas", "Glossário", "Vocabulário Controlado do Ministério da Saúde", "Bibliografia consultada" e "Descritores organizados por categorias".

Esta duplicação e remoção foi necessária uma vez que o código anteriormente desenvolvido marcou com "@" o nome das áreas temáticas, tanto no capítulo "Descritores organizados por categorias", como no capítulo "Áreas Temáticas da BVS Saúde Pública". Assim, por não ser possível distinguir que marcações estavam no capítulo das áreas temáticas e no capítulo dos descritores, achou-se pertinente apagar os capítulos acima referidos.

```

1 {
2   "AB": "Atenção Básica",
3   "ABEn": "Associação Brasileira de Enfermagem",
4   "ADT": "Assistência Domiciliar Terapêutica",
5   "AFE": "Autorização de Funcionamento de Empresa",
6   "AIDPI": "Atenção Integrada às Doenças Prevalentes na Infância",
7   "AIDS": "Síndrome da Imunodeficiência Adquirida",
8   "AIH": "Autorização de Internação Hospitalar",
9   "AIS": "Ações Integradas de Saúde",
10  "ANCD": "Associação Nacional de Centros de Defesa",
11  "ANS": "Agência Nacional de Saúde",
12  "ANVISA": "Agência Nacional de Vigilância Sanitária",
13  "APAC": "Autorização de Procedimentos de Alto Custo",
14  "APH": "Assistência Pré-Hospitalar",
15  "ASAJ": "Área de Saúde do Adolescente e do Jovem",
16  "BD-SIA/SUS": "Banco de Dados Nacional do Sistema de Informações Ambulatoriais do SUS",
17  "BLH": "Banco de Leite Humano",

```

Fig. 10: Documento JSON com siglas e respectivos significados.

Posto isto, foi criado outro ficheiro python, que abre e processa o ficheiro XML com as partes removidas, com código muito semelhante ao anterior (com exceção das partes de marcação da categoria, visto que estas marcações não eram pertinentes). A estrutura JSON definida para armazenar a informação extraída acerca das áreas temáticas foi a seguinte:

```

1 {
2   "Area 1": "Descricao da Area1",
3   "Area 2": "Descricao da Area2",
4   "Area 3": "Descricao da Area3",
5 }

```

Na Figura 11 pode verificar-se um excerto do documento JSON obtido.

```

{
  "Acidentes e Violência": "Refere-se ao conjunto de agravos à saúde que pode levar à óbito ou sequelas",
  "Administração e Planejamento em Saúde": "Refere-se à organização, elaboração de planos e políticas pu",
  "Alimentação e Nutrição": "Refere-se a todos os tipos de substâncias que têm por função alimentar ou r",
  "Ambiente e Saúde": "Refere-se ao estudo das interações entre os seres vivos e o meio, dedica-se a ana",
  "Atenção à Saúde": "Refere-se à proteção e atenção à saúde dos diversos grupos etários que corresponde",
  "Ciência e Tecnologia em Saúde": "Refere-se a investimentos públicos em ciência e tecnologia; desenvol",
  "Ciências Sociais em Saúde": "Refere-se aos estudos que se utilizam ou são elaborados pelas ciências s",
  "Comunicação em Saúde": "Refere-se ao conjunto dos meios de comunicação de massa voltados a divulgação",
  "Demografia": "Refere-se aos estudos das populações humanas, com o objetivo de caracterizá-las e anali",
  "Direito Sanitário": "Refere-se ao conjunto de leis e normas, nacional e internacional, que compõe o s",
  "Doenças Crônicas e Degenerativas": "Refere-se ao conjunto de doenças relacionadas a múltiplos fatores",
  "Doenças Infecciosas e Parasitárias": "Refere-se ao conjunto de infecções que podem ser adquiridas por",
  "Drogas de Uso Terapêutico e Social": "Refere-se aos efeitos causados pelo consumo de substâncias quim",
  "Economia da Saúde": "Refere-se aos estudos sobre gasto e financiamento em saúde, alocação e utilizaçã",
  "Epidemiologia": "Refere-se aos estudos retrospectivos e prospectivos da distribuição e dos determinan",
  "Equidade em Saúde e Social": "Refere-se à igualdade de recursos para necessidades iguais, de oportuni

```

Fig. 11: Documento JSON com as áreas temáticas e respectivas definições.

Foram extraídas, no total, 24 áreas temáticas do documento.

A mesma técnica foi utilizada para extrair os "Descritores organizados por categorias" (páginas 113 a 124, inclusive). Criou-se um novo ficheiro python e duplicou-se o documento XML original do qual se retiraram partes não relacionadas com o capítulo de interesse. O código desenvolvido para a extração da informação foi muito semelhante ao anterior.

A estrutura utilizada para esta informação foi a seguinte:

```
1 {  
2   "Categoria 1": ["Descritor 1", "Descritor 2", "Descritor 3"],  
3   "Categoria 2": ["Descritor 1"],  
4   "Categoria 3": ["Descritor 2", "Descritor 4"]  
5 }
```

Parte do resultado obtido está representado na Figura 12.

```
"Epidemiologia": [  
  "Vigilância sentinela",  
  "Vulnerabilidade"  
],  
"Equidade em saúde e social": [  
  "Centro Regional de Especialidade",  
  "Centros de saúde",  
  "Equidade",  
  "Perfil epidemiológico",  
  "Prevalência e regulação Assistencial",  
  "Universalidade"  
],  
"Ética e Bioética": [  
  "Bioética",  
  "Ética em pesquisa",  
  "Pesquisa em reprodução humana",  
  "Pesquisa em saúde",  
  "Pesquisa envolvendo seres humanos",  
  "Pesquisador responsável",  
  "Transplante de órgãos"  
],  
"História da Saúde Pública": [  
  "Franca explosão demográfica"  
],
```

Fig. 12: Documento JSON com os descritores e respectivas categorias.

Deste modo, foram extraídos, no total, 22 descritores.

4.2 Minidicionário de Cardiologista Autor Ricardo Silveira Mello.pdf

4.2.1 Análise do documento

O segundo documento diz respeito a um dicionário com a tradução de conceitos/expressões médicas de português para inglês e de inglês para português.

Assim, verificou-se que o documento apresenta 2 secções:

- Inglês-Português (páginas 10 a 30, inclusive): Esta informação está apresentada em duas colunas, da seguinte forma:

Termo em inglês - Tradução do termo para português

- Português-Inglês (páginas 32 a 51, inclusive)

Termo em português - Tradução do termo para inglês

No que diz respeito a exceções à regra, não foram detetados casos que se mostrassem diferentes dos demais.

Procedeu-se, então, para a análise do documento XML, obtido após a conversão do documento PDF, com o objetivo de encontrar padrões que permitissem a identificação rápida tanto dos termos, quer em inglês, quer em português, como das suas traduções. As conclusões retiradas desta análise foram:

- Os termos em português, que seriam traduzidos para inglês, foram convertidos com um tamanho de fonte igual a 13;
- Os termos em inglês, que seriam traduzidos para português, foram convertidos com um tamanho de fonte igual a 6 ou 7;
- As traduções, quer para português, quer para inglês, têm um tamanho de fonte igual a 8.

4.2.2 Processamento do documento

Após a análise do documento XML, o processamento e a extração de informação do documento foi relativamente fácil, quando comparado com o documento anterior.

Assim, tomou-se a decisão de marcar as traduções de português para inglês com a marca "@PI" e as traduções de inglês para português foram marcadas com "@IP". Para além disso, as traduções realizadas foram marcadas com @DEF. Sendo assim, fizeram-se as seguintes substituições:

```
1 # Marcacao dos termos em portugues que serao traduzidos para ingles
2 texto = re.sub(r"<.+font=\"13\"><b>(.)</b>.+\"", r"@PI\\1\\n", texto)
3
4 # Marcacao dos termos em ingles que serao traduzidos para portugueses
5 texto = re.sub(r"<.+font=\"[67]\"><b>(.)</b>.+\"", r"@IP\\1\\n", texto)
6
7 # Marcacao das traducoes
8 texto = re.sub(r"<.+font=\"8\">(.)<.+\"", r"@DEF\\1\\n", texto)
9
10 # Eliminacao de elementos HTML
11 texto = re.sub(r"<.+>", "", texto)
```

Após estas substituições, o documento XML resultante encontrava-se da seguinte forma:

```
@IPA SURMISE
@IPAORTIC KNOB
@IPA SURMISE (A CONJECTURE - SUSPI
@IPCION) (TO ASSUME ON SMALL EVI
@IPDENCE) -
@DEF Conjectura / Suposição
@IPA.C.L.S -
@DEFAdvanced Cardiovascular
@DEFLife Support
```

Fig. 13: Documento JSON após as substituições.

Pode verificar-se que as marcas, apesar de se referirem ao mesmo termo ou tradução, são colocadas repetidamente (como é o caso da definição "Advanced Cardiovascular Life Suport").

Para juntar as marcas foi implementado o seguinte ciclo *for*:

```
1 marcas = ["DEF", "IP", "PI"]
2
3 for marca in marcas: # ciclo para "combater" a greediness
4     pattern = rf"@{marca}(.*?) [\n]+?@{marca}(.+?)"
5
6     while (re.search(pattern, texto) != None):
7         # junta termos e definicoes com mais do que uma 1 marca
8         texto = re.sub(pattern, rf"@{marca}\1\2", texto)
```

Este ciclo foi fundamental para combater a *greediness*. Sem ele, se a substituição fosse apenas aplicada uma vez, não seriam abordados e substituídos todos os casos, já que haviam situações em que o mesmo termo tinha a mesma marca, 2 ou 3 vezes consecutivas.

O resultado deste ciclo *for* está representado na Figura 14

```
@IPA SURMISEAORTIC KNOBA SURMISE (A CONJECTURE - SUSPICION) (TO ASSUME ON SMALL EVIDENCE) -
@DEF Conjectura / Suposição
@IPA.C.L.S -
@DEFAdvanced Cardiovascular Life Support
```

Fig. 14: Documento JSON após o ciclo *for*.

Por fim, aplicou-se uma última substituição para remover os traços após os termos.

```
1 # remocao do - apos os termos
2 texto = re.sub(r"@(.+)\s+~\s+[\n]+", r"@1\n", texto)
```

Terminado o processamento do documento XML, foi utilizado novamente o findall para encontrar e agrupar as traduções.

```
1 traducoesIP = re.findall(r"@IP(.+)\n+@DEF(.+)", texto)
2 traducoesPI = re.findall(r"@PI(.+)\n+@DEF(.+)", texto)
```

Por fim, estas listas foram convertidas em dicionários diferentes cuja chave corresponde ao termo a ser traduzido e o valor corresponde à tradução.

4.2.3 Estrutura JSON

Para a informação extraída deste documento definiu-se apenas uma estrutura JSON:

```
1 {
2   "IP": {
3     "termo1 em ingles": "traducao do termo1 para portugues",
4     "termo2 em ingles": "traducao do termo2 para portugues"
5   },
6   "PI": {
7     "termo1 em portugues": "traducao do termo1 para ingles",
8     "termo2 em portugues": "traducao do termo2 para ingles"
9   }
10 }
```

Na Figura 15 pode verificar-se o documento JSON obtido.

```
"IP": {
  "WORKLOAD": "Carga de trabalho ",
  "WORTHY": "Que vale a pena / De valor / Valioso / Merecedor / Meritório/ Digno / Ilustre / Honrado",
  "WOUND": "Ferida / Ferimento",
  "WRIT": " Mandado / Ordenado"
},
"PI": {
  "1ª / 2ª / 3ª / 4ª / Bulha": " 1 / 2 / 3 / 4 / SOUND",
  "À esquerda / Ao contrário dos ponteiros do relógio / Sentido antihorário": "COUNTERCLOCKWISE",
  "A faculdade de fazer descobertas importantes e valiosas de maneira inesperada ou por acaso": "SERENDIPITY",
  "A investigação profunda sobre um assunto / Exame minucioso": "SCRUTINY",
  "A.C.L.S": "Advanced Cardiovascular Life Support",
  "A.E.D": "Automated External Defibrillator",
```

Fig. 15: Documento JSON com termos e respetivas traduções.

4.3 medicina.pdf

4.3.1 Análise do documento

O terceiro e último documento analisado revelou-se bastante rico em termos de informação, pelo que se procedeu ao tratamento de dados em formato xml. Durante a análise foram identificadas 3 secções de interesse, nomeadamente a tabela de abreviaturas, o vocabulário e os índices finais. Esta última acabou por não ser explorada, pois não introduzia informação nova, e poderia ser reconstruída com o conteúdo do vocabulário. Deste modo procedeu-se à separação da página relativa às abreviaturas (pág.17), uma vez que o seu formato era bastante diferente do restante documento, para o documento "abreviaturas.xml", mantendo-se o restante documento no ficheiro "medicina.xml". Posto isto, não se verificando mais informação relevante até à página 20, onde começa o dicionário em si, as páginas que a antecedem foram manualmente removidas.

Como última etapa de pré-processamento procedeu-se à remoção do conteúdo posterior ao final do dicionário (pág.544), pois a informação aqui contida remete para os índices.

As abreviaturas contidas na página 17 apresentam uma estrutura relativamente simples, sendo apenas diversos pares separados com espaçamento, tal como exemplo a seguir:

<i>Abreviatura</i>	<i>Conceito</i>
--------------------	-----------------

Relativamente ao dicionário em si, cada conceito é bastante flexível no conteúdo que apresenta. Considere-se o seguinte exemplo:

2279	hemisferio do cerebelo	m
	<i>Anatomía</i>	
es	hemisferio del cerebelo	
en	cerebellar hemisphere; hemisphere of cerebellum	
pt	hemisfério do cerebelo	
la	hemispherium cerebelli	

Fig. 16: Entrada 2279 do dicionário 'medicina.pdf'

Podemos verificar que uma entrada tem a si um número associado e a sua denominação galega, apresentando na maioria dos casos informação relativa à categoria gramatical. Esquematicamente:

Id numérico

Termo

Conceito em galego

Categoria gramatical

Informação relativa a género, se é símbolo ou sigla ou se é um substantivo com variabilidade de conjugação de género.

Cada entrada apresenta uma linha que corresponde à área em que o tema em questão se insere, seguida das respetivas traduções da denominação.

Área de aplicação

Traduções

Denominação em espanhol - *es*

Denominação em inglês - *en*

Denominação em português - *pt*

Denominação em latim* - *la*

*De salientar que a vasta maioria das entradas não apresenta esta última tradução. Dentro de cada conceito podemos ainda encontrar outros tipos suplementares de informação, nomeadamente:

SIN

Abreviatura que aparece após a área de aplicação e remete para sinónimos.

Nota

Nota relativa à denominação em questão, geralmente aparece como último elemento da entrada.

VAR

Abreviatura que remete para uma variante da denominação, aparece aos a área de aplicação ou de um sinónimo.

Por último, podemos ainda encontrar neste documento um outro tipo de entradas, chamadas pelos autores por 'entradas remissivas'. Estas entradas não se encontram numeradas

e tratam sinónimos ou variantes da entrada completa a que se refere, sendo a principal função dar a conhecer. Segue-se a estrutura:

Denominação

Conceito em galego

Vid.- conceito

A sigla 'Vid' significa 'Very important disease'

4.3.2 Processamento do documento

No tratamento do documento relativo às abreviaturas ('abreviaturas.xml'), retirou-se toda a informação relativa à formatação do ficheiro xml, ficando apenas conjuntos de abreviatura - conceito separados por linhas, dentro de tags text.

```
1 # Limpar tags de texto
2 doc = re.sub(r"<text\stop.*?>",r"<text>",doc)
3 #Retirar informacao de pagina
4 doc = re.sub(r'.+page.+\\n?', "",doc)
5 #Remocao de tudo ate ao titulo
6 doc = re.sub(r'.+\\n.+</b></text>\\n', "",doc)
```

Posto isto, procedeu-se à captura destes pares com 2 grupos de captura.

```
1 lista_abr_sig = re.findall(r'<text>(.)</text>\\n<text>(.)</text>',doc)
```

Passando agora à fração do documento correspondente ao dicionário em si. De modo a simplificar a interpretação do documento xml em mãos, começou-se por efetuar uma limpeza dos dados.

```
1 # Retirar informacao de formatacao das tags de texto
2 doc = re.sub(r"<text\stop.*?>",r"<text>",doc)
3 # Retirar espacamento antes do conteudo da tag
4 doc = re.sub(r"<text>\\s*</text>\\n",r"", doc)
5 # Retirar tags de inicio de pagina
6 doc = re.sub(r'<page number [\\w\\W]+?><text>\\d+</text>\\n', "",doc)
7 # Retirar tags de fim de pagina
8 doc = re.sub(r"</page>\\n", "",doc)
9 doc = re.sub(r"<fontspec.*\\n", "",doc)
```

Feita esta limpeza inicial procedeu-se à marcação de cada entrada do dicionário com @tb@ e ainda à marcação das traduções de cada denominação.

```
1 # Marcar o inicio de uma entrada
2 doc = re.sub(r"<text><b>\\s*(\\d.*)</b></text>",r"@tb@ \\1", doc)
3 # Marcar o inicio de entradas com 2 linhas
4 doc = re.sub(r"<text>\\s*(\\d+)\\s</text>\\n<text><b>(\\d.*)</b></text>",r"@tb@ \\1 \\2",doc)
5
6 # Marcar traducoes em espanhol
```

```

7 doc = re.sub(r"<text>\s*(es)+\s*</text>", "@es@", doc)
8 # Marcar traducoes em portugues
9 doc = re.sub(r"<text>\s*(pt)+\s*</text>", "@pt@", doc)
10 # Marcar traducoes em ingles
11 doc = re.sub(r"<text>\s*(en)+\s*</text>", "@en@", doc)
12 # Marcar traducoes em latim
13 doc = re.sub(r"<text>\s*(la)+\s*</text>", "@la@", doc)

```

Após nova análise verificou-se que durante a conversão de formato pdf para xml foram introduzidos alguns erros, nomeadamente a representação do separador ';' dentro de tags, quando separava palavras, a introdução de espaçamento e de quebras de linha. Verificou-se ainda que havia várias tags de texto com itálico (+19999), sendo apenas texto, pelo que se retiraram estas tags exceto quando uma outra tag se poderia encontrar dentro. De modo a corrigir isto desenvolveu-se as seguintes regex:

```

1 #extrair o conteudo de dentro das tags de italico, que nao tenha mais tags
2 doc = re.sub(r"<text><i>([-\\w\\s\\[\\]\\.]</i></text>", r"1", doc)
3 # retirar lixo nas linhas de ;
4 doc = re.sub(r"<text>\s*;\s*</text>", r";", doc)
5 # retirar tags vazias
6 doc = re.sub(r"<text><b>\s*</b></text>\n", "", doc)
7 # retirar linhas em branco entre palavras e ;
8 doc = re.sub(r"([\\.[\\]\\w]+)\n;\n", r"1;\n", doc)

```

Posto isto procedeu-se à marcação da categoria gramatical. Na maioria das situações esta categoria referia-se ao género do termo, denotando-se por f ou m, tendo a expressão regular sido alargada para lidar com s, sg e sb. Um problema que surgiu nesta abordagem foi numa conjugação de um termo na forma plural, fazendo com que hajam duas categorias associadas. Para lidar com esta situação criou-se uma outra expressão regular. É de notar que ao delimitar a categoria em ambos os lados, o nome do termo e as áreas de aplicação também ficam delimitadas.

```

1 # Retirar espacos e marcar categoria gramatical
2 doc = re.sub(r"s+([fmas]|sg|sb)\n\s*(\w)", r' #1#\n2', doc)
3 # Caso especial em que aparecia genero e pl (plural)
4 doc = re.sub(r's+([fma])\s*pl\n\s*([\w])', r' #1 pl#\n2', doc)
5 # Remocao de um erro de espacamento
6 doc = re.sub(r"s\n\s+", r" ", doc)

```

Seguidamente procedeu-se à marcação de sinónimos, linhas iniciadas por 'Vid.-' e ainda termos a negrito com a tag 'SUBT', que corresponderão às denominações das entradas remissivas. A abordagem tomada foi denotar apenas o início destas tags com '#', removendo-se sempre que possível erros introduzidos pela conversão, contudo é de notar que cada elemento fica limitado pelo começo de um outro, pelo que erros não capturados nestas expressões serão eliminados posteriormente.

```

1 # Marcacao de sinonimos
2 doc = re.sub(r'<text>\s*(SIN[\\w\\W]+?)</text>\n', r'#1\n', doc)
3 # Etiquetar VID.
4 doc = re.sub(r'<text>\s*(Vid.+)</text>', r'#1', doc)
5 # Lidar com situacoes que ficou em 2 linhas.

```

```

6 doc = re.sub(r'(#Vid.-\s\w+\s)\n<text>(.)</text>',r'\1\2',doc)
7 # Correcao para titulos multilinha.
8 doc = re.sub(r'(b@.)\n<text><i><b>(.)</b></i></text>\n.+<b>\s*(.)</b>.\n',
9             r'\1\2 #3#\n',doc)
10 # Marcacao de termos que aparecem a negrito.
11 doc = re.sub(r'<text><i><b>(.)</b></i></text>\n#',r'#SUBT \1\n',doc)
12 # Retirar linhas de ; pos italico.
13 doc = re.sub(r'<text><i>(.)</i></text>\n;',r'\1;',doc)

```

Uma vez que as tags de itálico que restavam delimitavam conteúdo de outras tags, este padrão foi retirado, por não ter mais utilidade. Procedeu-se então à correção de alguns erros encontrados inerentes à marcação de termos multilinha. A quarta e quinta expressão deste excerto são bastante semelhantes, contudo a última lida com situações em que não temos informação relativa à categoria gramatical, não captadas pela anterior. A junção destas expressões resulta num conjunto solução menor, pelo que se optou pela escrita em separado. Por último verificou-se que os elementos multilinha com tags bold não foram bem capturados, pelo que se procedeu às devidas correções.

```

1 # Retirar italico.
2 doc = re.sub(r'<text><i>(.)</i></text>',r'\1',doc) # retirar italico
3 # Formatacao de termos de 3 linhas e colocar tag no genero.
4 doc = re.sub(r'(b@.)\n.+<b>(.)</b>.\n.+<b>.*\s([mfa]).*</b>.\n',r'\1\2#\3#',
5             doc)
6 # Caso especifico de termos com 2 linhas, com genero e pl nao marcado
7 doc = re.sub(r'(b@.)\n<text><b>(.)\s([mfa])(\spl)</b></text>',r'\1\2 #3\4#',
8             doc)
9 # Tratar termos com 2 linhas e colocar tag no genero
10 doc = re.sub(r'(b@.)\n<text><b>(.)\s+([mfa])\s*</b></text>\n',r'\1\2#\3#\n',
11             doc)
12 # Escrever esta e a linha anterior nao estava a funcionar como esperado
13 doc = re.sub(r'(b@.)\n<text><b>(.)\s+([mfa]?)\s*</b></text>\n',r'\1\2#\3#\n',
14             doc)
15 # Ao analisar elementos com bold reparou-se que havia um subtítulo com 4
16   linhas, e varios com 2 ou 1. Nao se usou {n} para conseguir aceder aos
17   grupos de captura.
18 doc = re.sub(r'<text><b>(.)</b></text>\n?<text><b>(.)</b></text>\n?<text><b>
19   >(.)</b></text>\n?<text><b>(.)</b></text>\n?',r'#SUBT \1\2 \3\4\n',doc)
20 doc = re.sub(r'<text><b>(.)</b></text>\n?<text><b>(.)</b></text>\n',r'#SUBT
21   \1\2\n',doc)
22 doc = re.sub(r'<text><b>(.)</b></text>\n',r'#SUBT \1\n',doc)

```

Por fim etiquetou-se então o elemento 'VAR' (variante) e o elemento "Notas", tendo sido eliminadas as tags de texto sobrantes.

```

1 # Marcacao VAR
2 doc = re.sub(r'.+(VAR.)</text>',r'#\1',doc)
3 # Marcacao inicio de Notas
4 doc= re.sub(r'.+(Nota.)</text>\n',r'#\1\n',doc)
5 # Retirar tags de texts que faltam
6 doc = re.sub(r'<text>(.)</text>',r'\1',doc)

```

E procedeu-se à correção de erros que surgiram na análise do documento. A terceira expressão permite corrigir uma situação em que as marcações não ficaram como esperado, e como tal, quebravam as regras de captura que serão abordadas de seguida. Foi também este o caso com a 4ª expressão, na medida em que simplificaria a captura que houvesse um placeholder no espaço indicado.

```

1 # Caso específico em que CO2 quebra a regra de marcacao @tb@
2 doc = re.sub(r"(\#SUBT CO)\n@tb@s(2)",r"\1\2",doc)
3 # Erro de tag, duas seguidas referentes ao mesmo termo
4 doc = re.sub(r'(@tb@.+)\n@tb@s(.+)',r'\1\2',doc)
5 # Correcao de um erro introduzido anteriormente
6 doc = re.sub(r'(b@.+)\#\n\#SUBT\s([mf])\s+\n',r'\1 \2#\n',doc)
7 # Marcar os que nao tem genero
8 doc = re.sub(r'(b@.+)\#',r'\1Nap#',doc)
9 # Casos de 2 linhas com SUBT
10 doc = re.sub(r'(\#SUBT.+)\n\#SUBT(.+)\n',r'\1\2\n',doc)
11 doc = re.sub(r'(\#SUBT.+)\n(Vid.)',r'\1\2',doc)

```

Estando o documento num estado bastante mais simples que o seu estado inicial, deu-se como terminada esta etapa de processamento. Tal como já foi mencionado, cada entrada deste ficheiro foi marcada com @tb@, pelo que a expressão de captura irá apanhar tudo entre um b@ e o @t seguinte. Deste modo, é praticamente imediato que temos 2 problemas em mãos. Por um lado capturar a primeira entrada remissiva, que aparece antes do primeiro conceito, e por outro, capturar o último elemento, uma vez que, como nenhum lhe segue, ficaria em "aberto". A abordagem tomada foi incluir a primeira entrada remissiva no primeiro conceito, uma vez que será a este que diz respeito, e ainda acrescentar um @t no fim do documento para capturar o último conceito, resolvendo-se deste modo os problemas apresentados.

```

1 doc += '@t'
2
3 conceitos = re.findall(r"b@[\\w\\W]+?@t", doc)
4
5 sub_ini = "#SUBT A\\n\\#Vid.- adenina\\n@t" # este subtermo esta fora de todos os
        outros.
6 conceitos[0].strip("@t")
7 conceitos[0] += sub_ini

```

Tendo então uma lista de todos os conceitos, iterou-se sobre a mesma, tendo-se definido uma expressão regular para capturar as áreas de aplicação e uma outra para capturar os 3 diferentes grupos da primeira linha de cada entrada, o número, o título e a categoria gramatical. De seguida procedeu-se à captura das traduções.

```

1 dicionario = {}
2
3 for elem in conceitos:
4     tipos = re.search(r'b@.+\\n(.+)',elem).groups()
5     linha = re.search(r'b@s(\\d+)\\s(.+)\\s#(.+)#',elem) # lista de 1 tuplo
6     num, tit, gen = linha.groups()
7     tit = tit.strip()
8     tit = re.sub(r'\\s+', ' ', tit)
9     if gen=="Nap":

```

```

10     gen = ""
11     dicionario[num] = {"Termo":tit, "Categoria gramatical": gen, "Area(s) de
    aplicacao": [tipo for tipo in re.split(r'\s{2,}',tipos[0])], "Traducoes":
    {}}
12     trad= re.findall(r"(pt|en|es|la)@([\w\W]+?[@#])",elem)
13     for idioma in trad:
14         cod_pais = idioma[0]
15         tradu = idioma[1].replace("\n"," ")
16         tradu = tradu.replace("@","")
17         tradu= tradu.strip("#")
18         tradu= tradu.strip("@")
19         tradu= tradu.strip()
20         tradu = re.sub(r'\s+', ' ', tradu)
21         dicionario[num] ["Traducoes"] [cod_pais] = tradu

```

Por último foi necessário capturar os campos opcionais, sendo para isto determinante efetuar uma verificação da presença da tag desse campo na entrada. Em caso positivo, este campo é capturado e é lido retirado qualquer espaçamento ou marcação que não traduzisse informação relevante.

```

1 extras = ["#SIN", "#Nota", "#VAR"]
2     for extra in extras:
3         if extra in elem:
4             cont = re.search(r'{'([\w\W]+?)'[@#]'.format(extra),elem).groups()
5             extra = extra.strip("#")
6             cont = cont[0].strip("\n")
7             cont = cont.strip("-")
8             cont = cont.strip()
9             cont = re.sub(r'\s+', ' ', cont)
10            dicionario[num] [extra] = cont
11
12    if "#SUBT" in elem:
13        subt = re.findall(r'#SUBT(.+\n(.+\n)*?)#',elem)
14        vid = re.findall(r'#(Vid.+\n(.+\n)*?)'[@#]',elem) # um subt tem sempre
    um vid
15        rel = {}
16        for i in range(len(subt)):
17            subt_c = ""
18            vid_c = ""
19            for j in range(len(subt[i])):
20                subt_c += subt[i][j]
21            for k in range(len(vid[i])):
22                vid_c += vid[i][k]
23            subt_c = subt_c.strip("\n")
24            subt_c = subt_c.strip("-")
25            subt_c = subt_c.strip(" ")
26
27            vid_c = vid_c.strip("\n")
28            vid_c = vid_c.strip(" ")
29            rel[subt_c] = vid_c

```



```
30    dicionario[num]["Entrada remissiva"] = rel
```

4.3.3 Estrutura JSON

A estrutura json definida para o documento relativo às abreviaturas ("abrev.json") é relativamente simples, tal como o seu conteúdo, tendo sido definido da seguinte forma:

```
1 {  
2   "Abreviatura":  "Termo"  
3  
4 }
```

Deste documento foram extraídas 17 abreviaturas.

```
{  
  "a": "adjectivo",  
  "abrev": "abreviatura",  
  "arc": "forma arcaica",  
  "Br": "português do Brasil",  
  "col": "forma coloquial",  
  "cult": "forma culta",  
  "EE. UU.": "inglês americano",  
  "f": "substantivo feminino",  
  "lit": "forma literaria",  
  "loc": "locución",  
  "m": "substantivo masculino",  
  "pl": "plural",  
  "pop": "forma popular",  
  "Pt": "português de Portugal",  
  "s": "substantivo con variación de xénero",  
  "sb": "símbolo",  
  "sg": "sigla"  
}
```

Fig. 17: Ficheiro JSON com abreviaturas e respetivo termo

Gravitando este documento à volta do vocabulário médico e respetivas traduções, é natural que a esta parte corresponda uma estrutura mais complexa, como podemos de seguida ver, na estrutura completa de cada entrada deste ficheiro:

```

1 {
2   "Id numerico": {
3     "Termo" : "Conceito em galego.",
4     "Categoria gramatical": "Sigla relativa a categoria gramatical a
que o termo pertence.",
5     "Area(s) de aplicacao": [
6       "Area de aplicacao 1"
7     ],
8     "Traducoes": {
9       "es": "Conceito em espanhol",
10      "en": "Conceito em ingles",
11      "pt": "Conceito em portugues",
12      "la": "Conceito em latim"
13    },
14    "SIN": "Sinonimo",
15    "Nota": "Conteudo da nota",
16    "VAR": "Variante",
17    "Entrada(s) remissiva(s)": {
18      "Denominacao" : "Vid.- conceito"
19    }
20  }
21 }
22 }
```

Tal como esperado, o ficheiro "dados.json" resultante apresenta 5393 entradas.

```

{
  "509": {
    "Termo": "asistencia domiciliaria",
    "Categoria gramatical": "f",
    "Área(s) de aplicación": [
      "Organización sanitaria"
    ],
    "Traduções": {
      "es": "asistencia a domicilio; asistencia domiciliaria; atención domiciliaria",
      "en": "home care",
      "pt": "assistência domiciliar"
    },
    "SIN": "atención domiciliaria",
    "Nota": "Cfr. con \"hospitalización a domicilio\"",
    "VAR": "asistencia a domicilio",
    "Entrada remissiva": {
      "asistencia especializada": "Vid.- atención especializada",
      "asistencia médica": "Vid.- atención médica",
      "asistencia primaria": "Vid.- atención primaria"
    }
  }
},
```

Fig. 18: Entrada com todos os campos supracitados

5 Conclusão

Considera-se que todos os objetivos deste trabalho foram atingidos, tendo sido possível aplicar e aprofundar conhecimentos relativos a expressões regulares abordados durante as aulas de Processamento de Linguagem Natural em Engenharia Biomédica.

Em suma extraiu-se toda a informação que se considerou relevante dos ficheiros em formato PDF escolhidos, tendo esta sido guardada no formato JSON, de acordo com as estruturas definidas.

A principal dificuldade associada ao desenvolvimento deste trabalho foi, de uma forma geral, lidar com os casos específicos encontrados em cada um dos documentos.

Foi possível verificar que existem diferentes formas de extração de informação dos documentos, nomeadamente abordagens em que se realiza (ou não) uma limpeza dos documentos antes do seu processamento. Para além disso, foi possível compreender melhor e aprofundar a utilização de expressões regulares "greedy" e "lazy"