Universidade do Minho
Escola de Engenharia

Engenharia Biomédica
Universidade do Minho

Processamento de Linguagem Natural

MEBIOM – Informática Médica 2022/2023

Dicionário Médico

Relatório do Trabalho Prático 1

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 1 da unidade curricular de Processamento de Linguagem Natural.

Neste trabalho, foi proposta a aplicação do conhecimento adquirido em aula de maneira a processar diversos documentos PDF relativos a dicionários médicos.

O principal objetivo passa por analisar os documentos selecionados, de modo a conseguir extrair a informação considerada importante de cada um, para que possa ser utilizada em trabalhos futuros.

Assim, em primeiro lugar, foi feita uma análise dos diversos documentos disponíveis e foram selecionados três documentos, cuja informação contida em cada documento pareceu ser complementada pelos restantes. Posteriormente, procedeu-se à análise individual de cada documento, conversão para um tipo de ficheiro adequado, limpeza dos dados, criação de tags para realçar os elementos a extrair e, por fim, extração dos dados relevantes para uma estrutura definida. No fim, foi feita uma junção dos três ficheiros resultantes de modo a obter uma única estrutura que englobasse toda a informação extraída.

Os ficheiros selecionados foram: **Dicionário de Termos Médicos português-inglês-espanhol**, **Glossário de Termos Médicos Técnicos e Populares** e ***WIPO Pearl Covid-19 Glossary***.

Desenvolvimento

Dicionário de Termos Médicos português-inglês-espanhol

O documento Dicionário de Termos Médicos português-inglês-espanhol está dividido em 3 secções, nomeadamente a secção que começa com o termo inglês e apresenta as respetivas 2 traduções, a que começa com o termo espanhol e a que começa com o termo português, sendo que ambas as secções apresentam os mesmos termos. Por isso, para o presente trabalho foi escolhido trabalhar apenas a secção “portuguesa” do dicionário.

No começo, o documento PDF do dicionário foi passado para HTML, formato em que havia muito ruído que atrapalhava o seu processamento. Então, testou-se a conversão para o formato XML, formato que se mostrou mais limpo. Essa conversão foi feita com o auxílio do Ubuntu.

Processamento do ficheiro XML

Em Python, depois de aberto o ficheiro XML, foi feita uma limpeza de várias *tags* do documento até este ficar com apenas as *tags* ***bold*** nos termos e o restante sem *tags*.

Para isso, foram feitas as seguintes alterações:

- Obtenção do conteúdo entre as *tags* de texto;
- Eliminação de todo o conteúdo até ao primeiro termo da secção portuguesa do dicionário;
- Eliminação dos blocos de texto correspondentes à barra lateral português-inglês-espanhol;
- Eliminação dos números das páginas: mais tarde foi reconhecido um bug na conversão do ficheiro para XML, em que o primeiro termo de cada página era repetido depois do número da página, ou seja, no final mesmo antes da mudança de página; assim, no momento de eliminação do número da página foi apanhado também esse termo que estava duplicado;
- Eliminação das *tags* de página e especificações de formatação (*fontspec*);
- Eliminação das letras que identificam a secção do dicionário (A, B, C, …);
- Eliminação dos distintivos de feminino, masculino, plural, adj, adv, ... (incluídos numa *tag* de itálico);

A secção de código utilizada para tais alterações foi a seguinte, com as respetivas expressões regulares:

```
# apanhar tudo o que está entre tags de texto
text = re.sub(r'<text\b[^>]*>(.*?)</text>', r'\1', text)

# apagar todo o texto até começar o dicionário português-inglês-espanhol (1º termo)
index = text.find("<b>à morte (bras.)</b>")
if index != -1:
    text = text[index:]

# apagar blocos repetidos que correspondem à barra lateral português-inglês-espanhol
text = re.sub(r'<b>português<\/b>[\s\S]*?<b>espanhol<\/b>', '', text)

# apagar desde tag para número da página até fim de página (inclui termo que repete no inicio e fim da página - bug)
text = re.sub(r'<b>\d+(.*\n)+?<\/page>', '', text)

# apanhar as tags de inicio de página e fontspec
text = re.sub(r'<page.*(\n\s.*)?', '', text)
# apagar as letras que identificam as secções do dicionário (A, B, C,...)
text = re.sub(r'<b>[A-Z]</b>', '', text)
# apagar os distintivos de feminino, masculino, plural, adj, adv,...
text = re.sub(r'<i>\(?.*\)?</i>', '', text)
```

Figura 1 - Código para formatação inicial (limpeza) do documento.

Após limpeza inicial do documento, começou-se a formatar o resultado da limpeza até se conseguir chegar ao aspeto correto para a obtenção das entradas do dicionário. Para isso, foram feitas as seguintes alterações:

- Substituição de todas as sequências de \n por apenas um só;
- Obtenção de termos (designações) com mais do que uma linha e juntá-los numa só expressão termo;
- Obtenção de traduções com mais do que uma linha, incluindo casos em que a mudança de linha é depois de uma vírgula;
- Junção de palavras anteriormente separadas por "mudança de linha" (hífen)
    - Código utilizado para os 3 passos acima:

```
# apanhar termos (designações) com mais do que uma linha e juntá-los numa só expressão <b>termo</b>
text = re.sub(r'(<b>.+</b>\n)+',
              lambda match: f'<b>{" ".join(re.findall(r"<b>(.+?)</b>", match.group(0)))}</b>\n', text)

# apanhar traduções com mais do que uma linha, incluindo casos em que a mudança de linha é depois de uma vírgula
text = re.sub(r'([a-z]+,? ?.*)((?:\n[^A-Z<].*)+)',
              lambda match: match.group(1) + match.group(2).replace("\n", " "), text)

# juntar palavras separadas por "mudança de linha"
text = re.sub(r'-\s', r'', text)
```

Figura 2 - Código responsável pela junção de designações e traduções com mais de uma linha numa só.

- Eliminação de casos em que ficam () sem conteúdo no meio, porque teriam adj, adv, etc, conteúdo que foi eliminado nas *tags* de itálico, porém os parêntesis não faziam parte da *tag*, ou casos em que tem (adj.) no termo;
- Resolução de casos especiais onde a tradução estava partida de outra forma, por exemplo:
    - Por uma vírgula e mudança de linha, com a segunda linha a começar com letra maiúscula
    - Por mudança de linha, com a segunda linha a começar com letra maiúscula
- Resolução de casos individuais, como um termo repetido pelo mesmo bug mencionado anteriormente, mas que ficou antes e não depois do número da página (termo coçar)

Armazenamento da informação num dicionário guardado em formato *json*

Depois de todo o tratamento mencionado, foram obtidas e formatadas as *entries* do dicionário, através do código seguinte:

```
# ------------------------------------------------------------------------

# obtenção das entries do dicionário
entries = re.findall(
    r'<b>(.*)</b>\nU\n(.*)\nE\n(.*)', text)

# ------------------------------------------------------------------------

# formatação das entries
new_entries = [(designation, ({"en": description1, "es": description2}))
               for designation, description1, description2 in entries]

dic = dict(new_entries)

# ------------------------------------------------------------------------
```

Figura 3 - Obtenção e formatação das entries do dicionário.

O dicionário foi guardado num ficheiro *json*, para posteriormente ser trabalhado com os restantes dicionários escolhidos. O código para guardar o dicionário num ficheiro *json* e um excerto do ficheiro são apresentados nas imagens seguintes:

```
out = open('dicionario_obrigatorio.json', 'w', encoding='UTF-8')
json.dump(dic, out, ensure_ascii=False, indent=4)
out.close()
```

Figura 4 - Código para guardar dicionário num ficheiro json.

```
},
"acidente vascular cerebral (AVC)": {
    "en": "cerebrovascular accident, cerebral apoplexy",
    "es": "accidente vascular cerebral (AVC)"
},
"ácido desoxiribonucléico (DNA)": {
    "en": "deoxyribonucleic acid (DNA)",
    "es": "ácido desoxiribonucleico (ADN, DNA)"
},
"ácido fólico": {
    "en": "folic acid",
    "es": "ácido fólico"
```


Figura 5 - Excerto do dicionário guardado num ficheiro json.

Glossário de Termos Médicos Técnicos e Populares

O documento encontra-se organizado por ordem alfabética, em que cada entrada contém um termo e a correspondente expressão popular ou vice-versa.

Após algumas tentativas, foi decidido transformar o documento PDF em ficheiro HTML para proceder à sua análise. Na figura seguinte, apresenta-se um excerto do formato inicial do mesmo.


```
<i>à volta da boca</i> (pop) , <b>perioral</b> <br/>
<i>à volta da órbita</i> (pop) , <b>periorbital</b> <br/>
<i>à volta dos vasos sanguíneos</i> (pop) , <b>perivascular</b> <br/>
<i>abaixamento, abatimento, prostração</i> (pop) , <b>depressão</b> <br/>
<b>abcesso</b> , <i>abcesso, tumor</i> (pop) <br/>
<i>abcesso, tumor</i> (pop) , <b>abcesso</b> <br/>
<i>abcesso; acumulação de pus</i> (pop) , <b>empiema</b> <br/>
<b>abdómen</b> , <i>barriga, ventre</i> (pop) <br/>
```


Figura 6 - Excerto do ficheiro HTML gerado.

Como é possível visualizar na figura 7, o texto continha informação que não era pertinente para o resultado final e, de modo a facilitar as fases seguintes, foi feita uma limpeza do conteúdo. Foi também feita uma substituição de forma a juntar alguns fragmentos de texto que ficavam separados por quebras de página do ficheiro PDF original.

```
# retirar marcas não necessárias do texto
text = ''
for line in lines:
    line = re.sub(r' ', r' ', line)
    line = re.sub(r'</?a.*?>', r'', line)
    line = re.sub(r'[A-Z]?<br/>', r'', line)
    line = re.sub(r'<hr/>', r'', line)
    line = re.sub(r'[ ]+', r' ', line)
    line = re.sub(r'&#34;', r"'", line)
    text += line

# corrigir situações onde a expressão está separada
text = re.sub(r'\n\n\(', r'\(', text)
text = re.sub(r'<\/i>\s+<i>', r' ', text)
```

Figura 7 - Excerto de código que trata da limpeza do ficheiro.

Após os passos mencionados, o texto ficou com uma estrutura mais legível e fácil de trabalhar, como visto na figura 8.

```
<i>à volta da boca</i> (pop) , <b>perioral</b>
<i>à volta da órbita</i> (pop) , <b>periorbital</b>
<i>à volta dos vasos sanguíneos</i> (pop) , <b>perivascular</b>
<i>abaixamento, abatimento, prostração</i> (pop) , <b>depressão</b>
<b>abcesso</b> , <i>abcesso, tumor</i> (pop)
<i>abcesso, tumor</i> (pop) , <b>abcesso</b>
<i>abcesso; acumulação de pus</i> (pop) , <b>empiema</b>
<b>abdómen</b> , <i>barriga, ventre</i> (pop)
```

Figura 8 - Excerto do texto após limpeza do mesmo.

Como a ordem das entradas do documento não era sempre a mesma, ou seja, como poderia surgir primeiro o termo ou a expressão popular, foram consideradas duas fases de extração de dados: a primeira que selecionasse os casos em que primeiro aparecia a expressão popular e, depois, o termo; e uma segunda fase em que fossem captados os casos em que o termo surge em primeiro lugar, seguida da expressão popular.

Para ambos os casos, foi usada uma expressão regular semelhante, trocando apenas a ordem dos elementos. Para a primeira situação, foi usada a expressão regular `<i>(.*)<\/i>(?:.*)<b>(.+)<\/b>`, com três grupos de captura, estando o segundo silenciado. O primeiro grupo capta tudo o que se encontra dentro das tags HTML `<i></i>`, onde se encontra a expressão popular e o terceiro grupo capta o que se encontra dentro das tags `<b></b>`, onde estão incluídos os termos. A expressão regular usada na segunda situação foi `<b>(.+)<\/b>(?:.*)<i>(.*)<\/i>`, com a lógica oposta da explicada anteriormente.

```
# 1º capturar linhas que estão exp popular -> termo
entries1 = re.findall(r'<i>(.*)<\/i>(?:.*)<b>(.+)<\/b>', text)

# inverter o dicionario dado que estava ao contrário da ordem pretendida (termo -> ex popular)
dic = {termo2 : termo1 for termo1, termo2 in entries1}
```

Figura 9 - Excerto de código de seleção das linhas expressão popular -> termo.

```
# 2º capturar as linhas que estão termo - exp popular
entries2 = re.findall(r'<b>(.+)<\/b>(?:.*)<i>(.*)<\/i>', text)

# adicionar ao dicionario com os outros termos já captados antes
for termo, exp in entries2:
    dic[termo.strip()] = exp.strip()

# ordenar o dicionário por ordem alfabética
dic = sorted(dic.items())
```

Figura 10 - Excerto de código de seleção das linhas termo -> expressão popular.

O texto captado pelas expressões regulares foi guardado no dicionário, com 1819 entradas, em que a chave seria o termo e o valor seria a expressão regular. De notar que, no primeiro caso, foi necessário inverter a ordem dos grupos selecionados, visto que não se encontravam na ordem pretendida no dicionário.

Por fim, foi ordenado o dicionário final por ordem alfabética e foi guardado numa estrutura JSON, presente na figura 11.


```
{
    "'blister'": "frasco de X comprimidos recobertos de plástico",
    "(d)escamação": "formação excessiva de escamas na pele",
    "(herpes) zóster": "vírus instalado à volta das células sensitivas",
    "ACTH": "hormônio adreno-córticotrófico, corticotrofina",
    "Gram-negativo": "que não toma o corante de Gram",
    "Gram-positivo": "que toma o corante de Gram",
    "Petit mal; epilepsia menor": "pequeno mal, epilepsia com ataques pouco intensos",
    "abcesso": "abcesso, tumor",
    "abdominal": "ventral",
    "abdómen": "barriga, ventre",
```


Figura 11 - Dicionário com termos e expressões populares captados.

Notou-se que as entradas se encontram duplicadas, ou seja, o mesmo termo e respetiva expressão popular aparecem duas vezes no documento original: tanto na ordem termo → expressão, como na ordem expressão → termo. No entanto, achou-se que a estratégia implementada continuaria a ser a mais correta dado que salvaguardava casos onde tivessem ocorrido problemas na captação de algum elemento.

WIPO Pearl Covid-19 Glossary

O glossário WIPO Pearl Covid-19 Glossary encontra-se organizado por ordem alfabética, e a cada termo está associada uma descrição e tradução para alemão, espanhol, francês, japonês, coreano, português, russo e chinês.

Este documento foi fornecido em formato PDF e posteriormente convertido para diferentes formatos como TXT, HTML e XML, sendo que no fim optou-se pelo formato XML. Esta escolha deveu-se ao facto de este apresentar menos conteúdo irrelevante e o texto ser apresentado de forma mais clara e acessível. Na Figura 12, está representado o resultado obtido na conversão para XML.


```
WIPOPearl_COVID-19_Glossary.xml
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE pdf2xml SYSTEM "pdf2xml.dtd">

<pdf2xml producer="poppler" version="22.02.0">
<page number="1" position="absolute" top="0" left="0" height="1262" width="892">
    <fontspec id="0" size="36" family="ZDCDNY+HelveticaNeueLTStd-Bd" color="#000000"/>
<image top="132" left="0" width="895" height="750" src="output-1_1.jpg"/>
<text top="91" left="154" width="211" height="43" font="0"><b>WIPO Pearl  </b></text>
<text top="127" left="154" width="322" height="43" font="0"><b>COVID-19 Glossary</b></text>
</page>
<page number="2" position="absolute" top="0" left="0" height="1262" width="892">
</page>
<page number="3" position="absolute" top="0" left="0" height="1262" width="892">
<text top="91" left="163" width="211" height="43" font="0"><b>WIPO Pearl  </b></text>
<text top="127" left="163" width="322" height="43" font="0"><b>COVID-19 Glossary</b></text>
</page>
<page number="4" position="absolute" top="0" left="0" height="1262" width="892">
```


Figura 12 - Excerto do ficheiro XML obtido

De seguida, retirou-se as *tags* que não eram importantes como por exemplo, <text>, <page>, <i> e <fontespec> e eliminou-se todo o texto até a página em que começavam os termos.

```
1
2
3   <b>acute respiratory distress syndrome</b>
4
5   (syn.)
6   ARDS
7   Respiratory disease characterized by the rapid onset
8   of widespread inflammation in the lungs.
9   MEDI, Pathology
10  <b>    AR</b>
11  ةمزلاتم
```

Figura 13 - Excerto do texto após a limpeza de dados

Posteriormente, removeu-se os espaços do texto e as traduções que não pretendíamos adicionar ao dicionário, mais precisamente a tradução alemã, espanhola, japonesa, coreana, russa e chinesa. Como resultado o texto passou a ser composto pelos termos, descrição, tradução portuguesa e francesa, como se pode observar no excerto da Figura 14.

```
1   <b>acute respiratory distress syndrome</b> (syn.) ARDS  Respiratory disease characterized by the rapid onset of
    widespread inflammation in the lungs. MEDI, Pathology <b>FR</b>  syndrome de détresse respiratoire aiguë, (syn.)
    SDRA  <b>PT </b>  síndrome do desconforto respiratório agudo, (syn.) SDRA  <b>adaptive immunity</b> (syn.)
    acquired immunity  Immunity acquired during a person?s lifetime upon exposure to an infectious agent either by
    previous infection or immunization. MEDI, Anatomy &amp;amp; physiology <b>FR</b>  immunité acquise, (syn.) immunité
    adaptative  <b>PT</b>  imunidade adquirida, (syn.) imunidade adaptativa  <b>aerosol</b> Suspension of solid or
```

Figura 14 - Remoção dos espaços e tradução

Uma vez que nem todos os termos e traduções tinham sinónimos e que em alguns casos esses eram compostos pelas iniciais do termo decidiu-se eliminar os sinónimos. Para isso utilizou-se a expressão regular apresentada na Figura 15 que permitiu eliminar de uma só vez as três situações em que estes apareciam.

```
# eliminar os sinónimos quer nos termos quer nas descrições/traduções
# utilização de OU lógico "|"
lines = re.sub(r',?\s?\(syn\.\).*?\s+([A-Z][a-z]|<b>)', r'\1', lines)
```

Figura 15 - Excerto do código para eliminar sinónimos

Estas situações podem ser observadas na Figura 16 e correspondem aos casos em que:

- (syn.) está seguido por uma palavra composta unicamente por maiúsculas seguida da descrição;
- (syn.) está seguido por algumas palavras que apenas contêm minúsculas e que são seguidas da descrição;
- (syn.) tem uma vírgula antes e depois tem o sinonimo da tradução.

```
1   <b>acute respiratory distress syndrome</b> (syn.) ARDS  Respiratory disease characterized by the rapid onset of widespread inflammation
    in the lungs. MEDI, Pathology <b>FR</b>  syndrome de détresse respiratoire aiguë, (syn.) SDRA  <b>PT </b>  síndrome do desconforto
    respiratório agudo, (syn.) SDRA  <b>adaptive immunity</b> (syn.) acquired immunity  Immunity acquired during a person?s lifetime upon
    exposure to an infectious agent either by previous infection or immunization. MEDI, Anatomy &amp;amp; physiology <b>FR</b>  immunité
    acquise, (syn.) immunité adaptative  <b>PT</b>  imunidade adquirida, (syn.) imunidade adaptativa  <b>aerosol</b> Suspension of solid
    or liquid particles in a gas. CHEM, Chemical elements &amp;amp; compounds <b>FR</b>  aérosol  <b>PT</b>  aerossol  <b>ageusia</b>
```

Figura 16 – Excerto do texto antes do tratamento de dados

Após o a tratamento dos dados procedeu-se a captura do termo, descrição e traduções restantes para que pudessem ser adicionados a um dicionário. Para isso, utilizou-se a expressão regular \s*(.*?)\s*<\/b>(.*?)FR\s*<\/b>\s*(.*?)\s*PT\s*<\/b>\s*(.*?)(?=|\Z) composta por 4 grupos de captura:

- O primeiro grupo captura qualquer sequência dentro das *tags* HTML e e corresponde ao termo.
- O segundo grupo captura qualquer sequência que aparece entre o primeiro grupo de captura e a *tag* HTML FR e corresponde à descrição.
- O terceiro grupo corresponde a tradução francesa e captura qualquer sequência que estiver entre a *tag* HTML FR e a *tag* PT.
- Por último, o quarto grupo corresponde a tradução portuguesa e captura o que estiver entre a *tag* PT e a próxima *tag* ou até ao final do texto se não houver mais *tags* .

```
# Obtenção dos tuplos para posterior criação do dicionário
entries = re.findall(r"<b>\s*(.*?)\s*<\/b>(.*?)<b>FR\s*<\/b>\s*(.*?)\s*<b>PT\s*<\/b>\s*(.*?)(?=<b>|\Z)", lines, re.DOTALL)

entries = [(term.strip(), desc.strip(), es.strip().rstrip(), pt.strip().rstrip()) for term, desc, es, pt in entries]


# Formatação das entries para o dicionário
new_entries = [(designation, ({"desc": description1, "fr": description2, "pt": description3}))
               for designation, description1, description2, description3 in entries]

dic = dict(new_entries)
```

Figura 17 - Excerto do código de obtenção dos túpulos

Uma vez que nos outros dicionários escolhidos o termo e a descrição estavam em português decidiu-se traduzir a descrição para que quando este dicionário fosse adicionado ao dicionário conjunto mantivesse a estrutura.

```
for key, value in dic.items():
    dic[key] = {"desc_pt" : GoogleTranslator(source='en', target='pt').translate(value['desc']),
                "desc_en" : value['desc'],
                "pt" : value['pt'],
                "fr" : value['fr']}
```

Figura 18 - Excerto do código para traduzir a descrição

Por fim, o texto capturado foi guardado num dicionário com uma estrutura JSON, como pode ser observado na Figura 19.

```
"aerosol": {
    "desc_pt": "Suspensão de partículas sólidas ou líquidas em um gás.",
    "desc_en": "Suspension of solid or liquid particles in a gas.",
    "pt": "aerossol",
    "fr": "aérosol"
},
"ageusia": {
    "desc_pt": "Distúrbio caracterizado pela perda do paladar.",
    "desc_en": "Disorder characterized by loss of taste.",
    "pt": "ageusia",
    "fr": "agueusie"
},
```

Figura 19 – Dicionário com termos, descrições e traduções

Junção de todos os documentos

Para a junção de todos os documentos trabalhados num só, começou-se por avaliar as interseções dos dicionários dois a dois, obtendo-se os seguintes resultados:

- Interseção Dicionário Termos Médicos com Glossário de Expressões Populares
  - 635 termos comuns

- Interseção Glossário de Termos COVID com Glossário de Expressões Populares
  - 16 termos comuns

- Interseção Dicionário Termos Médicos com Glossário de Termos COVID
  - 23 termos comuns

- Interseção dos 3 documentos
  - 8 termos comuns -> muito fraca interseção

Assim, tendo em conta as interseções encontradas, o que se achou razoável fazer, dado que o Glossário de Termos COVID tem muito pouca compatibilidade com os restantes, foi juntar à interseção que se mostrou mais significativa, com 635 termos, os termos não comuns do Glossário COVID. Dessa forma, a união final dos documentos foi a seguinte:

(*Glossário Exp. Populares* ∩ *Dicionário Termos Médicos PT_EN_ES*) ∪ *Glossário COVID*

Como exemplo, o código desenvolvido para a obtenção do dicionário relativo à expressão anterior foi o seguinte, que nos temos comuns aos 3 dicionários coloca toda a informação dos 3, adiciona os termos COVID não comuns e ainda adiciona todos os termos da interseção dos 2 primeiros:

```python
# união do dicionário de covid com dicionario de exp populares + en_pt
dic_final = {}
count = 0
for key, value in covid.items():
    if key in exps_enpt.keys():
        count += 1
        dic_final[key] = {"desc_pt" : value['desc_pt'],
                          "exp pop" : exps_enpt[key]['exp pop'],
                          "desc_en" : value['desc_en'],
                          "en" : exps_enpt[key]['en'],
                          "es" : exps_enpt[key]['es'],
                          "fr" : value['fr']}
    else:
        count += 1
        dic_final[key] = {"desc_pt" : value['desc_pt'],
                                      "en" : value['en'],
                                      "desc_en" : value['desc_en'],
                                      "fr" : value['fr']}

for key, value in exps_enpt.items():
    if key not in dic_final.keys():
        count += 1
        dic_final[key] = value
```

Figura 20 - Código para a obtenção do dicionário final.

Dessa união resultou um dicionário final guardado num ficheiro *json* com **773 termos**.

Conclusão

O desenvolvimento deste trabalho permitiu consolidar os nossos conhecimentos relativamente ao processamento de linguagem natural através de expressões regulares.

Na sua entrega, consideram-se os objetivos propostos para este trabalho concluídos, uma vez que se conseguiu criar um dicionário composto pelos resultados da análise de três outros dicionários tudo isto através do uso de expressões regulares.

Futuramente, para a melhoria deste trabalho deveriam ser analisados mais dicionários que tivessem um maior grau de compatibilidade com o dicionário de termos médicos português-inglês-espanhol e com o glossário de termos médicos técnicos e populares.