Universidade do Minho
Mestrado em Informática Médica

Natural Language Processing
Assignment 1

Unidade Curricular: Processamento de Linguagem Natural em Eng. Biomédica

Ano Letivo: 2022/2023

Trabalho realizado por:

Maria da Conceição Vieira Mota, PG51210

Susana Isabel Pereira Martins, A93790

Índice

1. Introdução

O processamento de linguagem natural (PLN) é uma ferramenta valiosa em muitas áreas da ciência, incluindo a engenharia biomédica. A capacidade de analisar grandes quantidades de dados não estruturados, como textos e documentos, é crucial para descobrir novos conhecimentos e insights num campo tão complexo como a biomedicina. No entanto, a complexidade dos dados textuais em documentos biomédicos pode tornar difícil a extração de informações relevantes.

Para lidar com essa complexidade, as expressões regulares têm se mostrado uma ferramenta poderosa e eficiente no processamento de dados textuais. As expressões regulares permitem identificar e extrair padrões específicos de texto em grandes conjuntos de dados, permitindo que informações importantes sejam extraídas de maneira rápida e eficiente.

2. Contexto e Requisitos

Este trabalho prático tem como objetivo a aplicação dos conhecimentos obtidos nas aulas para processamento de documentos PDF.

Para isso, foi proposto a implementação deste projeto com o objetivo de recuperar informações dos ficheiros fornecidos e armazenar essas informações para que possam ser utilizadas em trabalhos futuros.

Os requisitos incluem a definição de analisadores para extrair informações relevantes de cada documento PDF e que o resultado final seja uma estrutura contendo os dados considerados relevantes a serem preservados em, por exemplo, num ficheiro JSON, sendo recomendadas as seguidas etapas:

1. Análise dos documentos PDF, seleção das informações relevantes;
2. Definição de uma estrutura para representar a estrutura de dados a ser extraída;
3. Conversão dos documentos em formato PDF para um formato conveniente para sua manipulação (xml ou texto);
4. Limpeza dos dados, removendo elementos desnecessários;
5. Extração de campos relevantes nas estruturas de dados previamente definida e armazenamento dessa informação num ficheiro em formato JSON.

O processamento do ficheiro dicionario_termos_medicos_pt_es_en.pdf é **obrigatório**.

3. Abordagem e Implementação

Dado o desafio de definir uma estrutura para representar a informação a ser extraída na fase inicial, optamos por dividir a implementação deste trabalho prático em três fases principais: análise e extração, correlação da informação e definição da estrutura final e armazenamento dos dados.

3.1. Fase 1: Análise e extração de informação

Como mencionado anteriormente, encontramos dificuldades em identificar como relacionar os ficheiros e extrair a informação mais relevante para a criação de uma estrutura de dados. Nesse sentido, iniciamos o processo de análise e extração de informação em cada ficheiro, procurando identificar a informação mais relevante e padrões que pudessem ser tratados com expressões regulares.

Inicialmente, os ficheiros PDF foram convertidos para os formatos de texto, XML e HTML utilizando os comandos 'pdftotext', 'pdftohtml -xml' e 'pdftohtml', respetivamente. Antes do processamento de cada ficheiro, tentamos analisar o qual formato mais adequado para extrair a maior quantidade de informação possível. A informação extraída de cada ficheiro é armazenada numa estrutura em json que consideramos de fácil acesso e manipulação.

De seguida, serão apresentados mais detalhes sobre a informação extraída e as principais dificuldades encontradas no processo de extração.

1. dicionario_termos_medicos_pt_es_en.pdf

Na análise deste documento identificamos que ele é composto por três dicionários de termos médicos:

- english – spanish – portuguese
- español – inglés – portugués
- português – inglês – espanhol

Inicialmente começamos por fazer o processamento usando o formato de texto tentando usar um padrão como, por exemplo “termo E tradução P tradução”, mas identificamos vários problemas quando existem quebras de linhas e não conseguimos identificar um padrão que conseguisse distinguir as quebras no meio das traduções, pelo resolvemos analisar o formato XML. Neste formato, conseguimos identificar que este seguia mais ou o padrão:

```
"""<text[^>]*><b>(.+?)</b></text>
    \s?
    <text[^>]*>E</text>
    \s?
    <text[^>]*>(.*?)</text>
    \s?
    <text[^>]*>P</text>
    \s?
    <text[^>]*>(.*?)</text>
"""
```

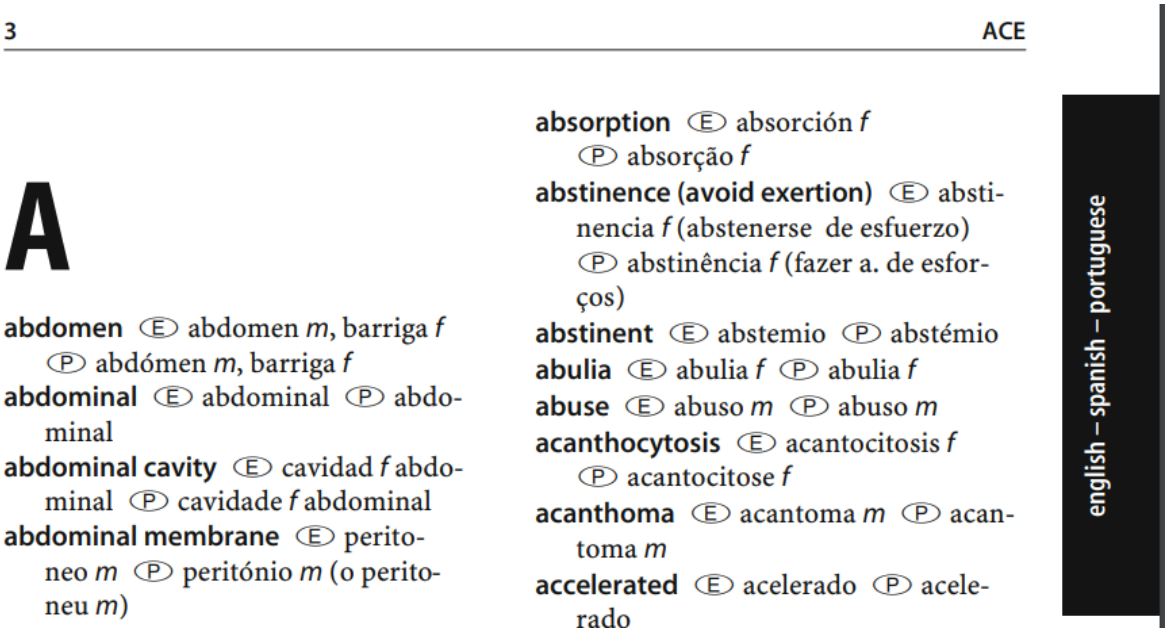
Neste formato também identificamos alguns problemas como headers no meio dos termos e das respetivas traduções e quebras de linhas, principalmente no meio de palavras. Para a resolução do problema dos headers, tivemos de identificar e remover essas linhas para não interferir com a captura de texto nas quebras de linhas.

Para isso, identificamos que, neste caso, o header aparecia sempre na posição onde o top era igual a 104, e por isso utilizamos a seguinte expressão regular para excluir estas linhas.

```
r'<text top="104"[^>]* height="50"[^>]*><b>[A-Z]</b></text>\s?'
```

Relativamente às quebras de linhas no meio das palavras, falaremos com mais detalhe no processamento do próximo ficheiro.

Como resultado para o PDF com o formato:



3 ACE

A

abdomen Ⓔ abdomen *m*, barriga *f* Ⓟ abdómen *m*, barriga *f*
abdominal Ⓔ abdominal Ⓟ abdominal
abdominal cavity Ⓔ cavidad *f* abdominal Ⓟ cavidade *f* abdominal
abdominal membrane Ⓔ peritoneo *m* Ⓟ peritónio *m* (o peritoneu *m*)
absorption Ⓔ absorción *f* Ⓟ absorção *f*
abstinence (avoid exertion) Ⓔ abstinencia *f* (abstenerse de esfuerzo) Ⓟ abstinência *f* (fazer a. de esforços)
abstinent Ⓔ abstemio Ⓟ abstémio
abulia Ⓔ abulia *f* Ⓟ abulia *f*
abuse Ⓔ abuso *m* Ⓟ abuso *m*
acanthocytosis Ⓔ acantocitosis *f* Ⓟ acantocitose *f*
acanthoma Ⓔ acantoma *m* Ⓟ acantoma *m*
accelerated Ⓔ acelerado Ⓟ acelerado

english – spanish – portuguese

Geramos três ficheiros, um para cada dicionário, com o seguinte output:

- dicionario_termos_medicos_en_es_pt.json


```
{
  "abdomen": {
    "es": "abdomen m, barriga f",
    "pt": "abdómen m, barriga f"
  },
  "abdominal": {
    "es": "abdominal",
    "pt": "abdominal"
  },
```


- dicionario_termos_medicos_es_en_pt.json


```
{
  "a la muerte": {
    "en": "mortally ill, dying",
    "pt": "à morte (bras.)"
  },
  "abdomen m": {
    "en": "belly, abdomen",
    "pt": "abdómen m"
  },
```


- dicionario_termos_medicos_pt_en_es.json


```
{
  "à morte (bras.)": {
    "en": "mortally ill, dying",
    "es": "a la muerte"
  },
  "abaulamento m": {
    "en": "swell",
    "es": "hinchazón f"
  },
```


2. Dicionario_de_termos_medicos_e_de_enfermagem.pdf

Neste documento, a informação que consideramos mais relevante é composta por um termo e respetiva descrição, no entanto, esta informação está disposta por colunas no PDF original. Ao analisar a conversão deste PDF para o formato de texto verificamos que a definição do termos ficava sobreposta, misturada sendo impossível distinguir os termos da

respetiva descrição, pelo que a utilização deste formato para a extração de informação ficou fora de questão.

Nesse sentido, consideramos novamente que o melhor formato seria o de XML, dado que, neste ficheiro as zonas que queremos extrair a informação está em colunas no ficheiro PDF, faz com que seja impossível usar o formato de texto, pois no caso, a informação ficava toda misturada sendo impossível distinguir os termos da respetiva descrição.

Novamente neste caso, identificamos problemas com footers, headers e numeração da página, sendo necessário a expressões regulares novamente com padrões na posição em que apareciam, como o top e muitas vezes sendo necessário ser mais específico e especificar também a altura e fonte em que este tipo de informação aparecia.

Outro dos problemas, foi novamente a dificuldade com as quebras de linhas, principalmente a meio das palavras. Por exemplo:

A palavra “sífilis” aparece no XML em três linhas diferentes e na última linha com um espaço.

```
<text top="531" left="104" width="105" height="13" font="11">de revestimento; a sí</text>
<text top="531" left="209" width="7" height="13" font="12">fi</text>
<text top="531" left="212" width="71" height="13" font="11"> lis, quando a </text>
```

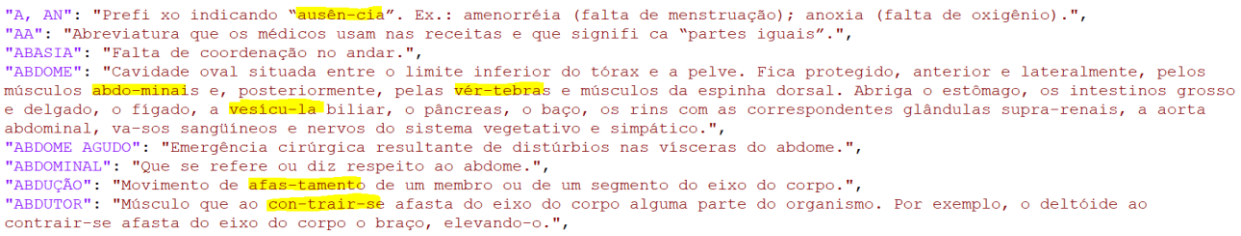
Ao construir a frase final existia espaço no meio da palavra.

```
radiação do elemento rádio. E, ainda, anomalias da constituição no desenvolvimento do útero, infl amação de seus tecidos de
revestimento; a sífi lis, quando a gestação, em geral, é interrompida no 5o mês.",
```

Outra situação identificada é o facto de existirem muitas palavras com um ‘-’ traço para identificar a quebra de linha na coluna, por exemplo, no PDF aparece da seguinte forma.

A, AN - Prefixo indicando “ausên-
cia”. Ex.: amenorréia (falta de
menstruação); anoxia (falta de
oxigênio).

Como resultamos tínhamos inúmeras palavras cortadas:

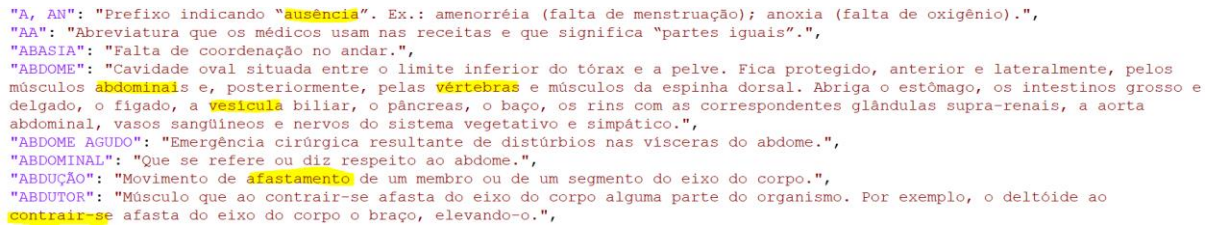


Para resolver esta situação verificamos que poderíamos considerar que sempre que existisse um "-" imediatamente antes do fecho da tag "-</text>" e um espaço no inicio da tag "> " estes poderia ser removidos.

Para isso, usamos a seguinte expressão regular que exclui o '-':

```
text = re.findall('<text [^>]*>\s?(.*?)-?</text>', text)
```

Desta forma conseguimos extrair o conteúdo do texto e obter o resultado pretendido:

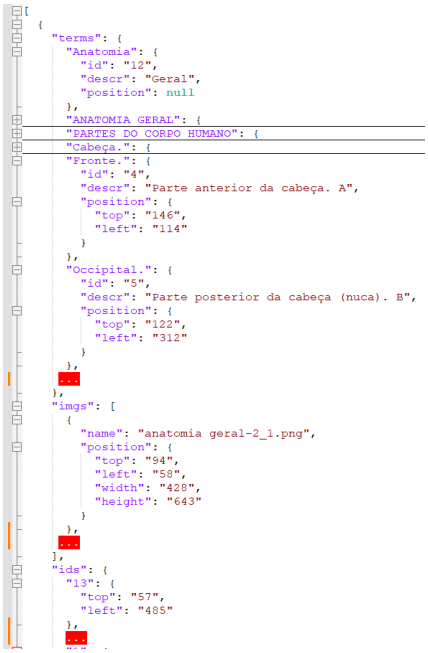


3. anatomia geral.pdf

Este documento é relativo a termos onde a sua maioria está identificado ou relacionado nas imagens que aparecem no PDF. A principal dificuldade foi perceber que de que forma poderíamos guardar esta informação de forma que conseguíssemos reconstruir o relacionamento com as imagens e os termos. Então verificamos que por padrão tínhamos páginas com termos associados a identificadores e de seguida uma página com as imagens e respetivos identificadores.

De forma que fosse possível reconstruir a informação consideramos importante guardar os termos com o respetivo identificador e a posição (top e left) em que esse identificador aparece na imagem. No entanto, identificámos que nem todos os termos estão relacionados nas imagens. Para estes casos, apenas temos uma seção com a lista de termos relacionado com as imagens, mas sem posição associada.

Como resultado para o PDF de anatomia, geramos um ficheiro com a seguinte estrutura, uma lista onde cada elemento da lista é um dicionário contendo termos (e respetivos identificadores, descrição e posição quando identificada), uma lista de imagens associadas a estes termos (com as respetivas posições) e um dicionário com todos os ids identificados na página e respetiva posição. Estes ids acabam por ser informação repetida com a posição de cada termo, posteriormente poderia ser removida para ocupar menos espaço.



4. CIH Bilingual Medical Glossary English-Spanish.pdf

Este documento é um PDF com um glossário, tendo uma tabela com termo e respetiva designação. Novamente aqui, no formato de texto a informação não aparecia sequencial pelo que escolhemos mais uma vez o formato xml. Mas, desta vez não estávamos a encontrar nenhum padrão para distinguir os termos das designações e contemplar possíveis quebras de linhas que misturasse a informação, até que, percebemos que o atributo left era o único padrão que nos permitia identificar de forma distinta cada um deles.

Por exemplo, para:

Doctor/Physician	Médico, Doctor
Attending Physician	Médico de Planta
Intern	Interno, residente de primer año

O termo aparece sempre com left=”108” e a descrição com left=”486”.

```
<text top="463" left="108" width="121" height="16" font="4">Doctor/Physician  </text>
<text top="463" left="270" width="4" height="16" font="4"> </text>
<text top="463" left="324" width="4" height="16" font="4"> </text>
<text top="463" left="378" width="4" height="16" font="4"> </text>
<text top="463" left="432" width="4" height="16" font="4"> </text>
<text top="463" left="486" width="105" height="16" font="4">Médico, Doctor </text>
<text top="484" left="108" width="136" height="16" font="4">Attending Physician </text>
<text top="484" left="270" width="4" height="16" font="4"> </text>
<text top="484" left="324" width="4" height="16" font="4"> </text>
<text top="484" left="378" width="4" height="16" font="4"> </text>
<text top="484" left="432" width="4" height="16" font="4"> </text>
<text top="484" left="486" width="120" height="16" font="4">Médico de Planta </text>
<text top="505" left="108" width="58" height="16" font="4">Intern   </text>
<text top="505" left="216" width="4" height="16" font="4"> </text>
<text top="505" left="270" width="4" height="16" font="4"> </text>
<text top="505" left="324" width="4" height="16" font="4"> </text>
<text top="505" left="378" width="4" height="16" font="4"> </text>
<text top="505" left="432" width="4" height="16" font="4"> </text>
<text top="505" left="486" width="217" height="16" font="4">Interno, residente de primer año </text>
```

Apenas desta forma conseguimos distinguir os termos das descrições e contemplar quebras de linhas.

Neste documento existe ainda mais três seções com prefixos, sufixos e roots, neste caso, temos uma tabela em dois, portanto o prefixo tem uma posição no left=”108” quando está à esquerda e um left=”460” quando está à direita, a mesma coisa para o significado.

Prefix	Meaning	Prefix	Meaning
A,an-	*Without, not, lack of*	*Ab-*	*Away from*
Ad-	*Toward*	*Ante-*	*Before, forward*
Anti-	*Against*	*Auto-*	*Self, own*

```
<text top="526" left="108" width="39" height="18" font="10"><i>A,an-</i></text>
<text top="545" left="108" width="28" height="18" font="10"><i>an- </i></text>
<text top="526" left="184" width="149" height="18" font="10"><i>Without, not, lack of </i></text>
<text top="545" left="184" width="5" height="18" font="10"><i> </i></text>
<text top="526" left="460" width="30" height="18" font="10"><i>Ab- </i></text>
<text top="526" left="528" width="82" height="18" font="10"><i>Away from </i></text>
<text top="544" left="108" width="30" height="18" font="10"><i>Ad- </i></text>
<text top="544" left="184" width="59" height="18" font="10"><i>Toward </i></text>
<text top="544" left="460" width="44" height="18" font="10"><i>Ante- </i></text>
<text top="544" left="528" width="117" height="18" font="10"><i>Before, forward </i></text>
```

Neste documento foi fundamental a utilização de padrões identificados nos atributos das tags <text>, caso contrário não teríamos como distinguir a informação.

Como resultado geramos um ficheiro com o glossário e outro com os prefixos, sufixos e roots.

5. Glossário de Termos Médicos Técnicos e Populares.pdf

Para este glossário de termos médicos utilizamos o ficheiro de texto dado que o padrão é constante, sem quebras de linhas, por exemplo:

“a milionésima parte de um grama (pop), micrograma”

Para extrair o termos e respetiva explicação foi definida a seguinte expressão regular:

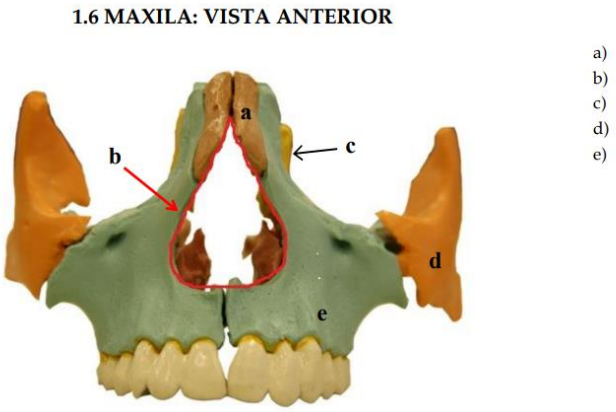
```
pattern = r'^(.*?)\s*\((pop)\)\s*,\s*(.*?)$'
```

Como resultado é gerado um ficheiro JSON com os termos e respetivas explicações:


```
{
    "'blister'": {
        "pop": "frasco de X comprimidos recobertos de plástico"
    },
    "(d)escamação": {
        "pop": "formação excessiva de escamas na pele"
    },
    "(herpes) zóster": {
        "pop": "virus instalado à volta das células sensitivas"
    }
```

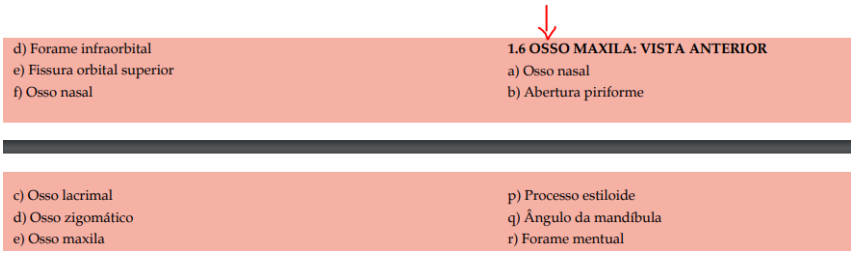

6. ossos.pdf

O documento ossos.pdf é composto essencialmente por páginas com um titulo e com imagens com letras em determinas posições.



No final do documento aparecem novamente os títulos de cada um dos capítulos com as letras e respetivas dignações.

Nesse sentido, procuramos extrair de cada página as imagens, as letras identificadas nas imagens e respetivas posições (top e left) e no final identificar esses títulos e relacioná-los com as respetivas imagens. O grande obstáculo neste caso foi que muitas vezes o título que aparece na imagem não é exactamente igual ao que aparece nas designações. Logo no primeiro capitulo temos “1.1 CRÂNIO: VISTA ANTERIOR - I” e na designação “1.1**.** CRÂNIO: VISTA ANTERIOR - I” com um ponto no meio, o que fazia logo com que não conseguíssemos fazer a correspondência. Outros casos onde numa seção tinha no final do titulo “ – I” ou “ – II” e depois na outra secção não tinha. Para estes casos implementamos uma função para uniformizar e retirar o último “.” na numeração e o titulo “ – I” ou “ – II” no final do titulo. No entanto entramos casos mais complicados que não conseguimos fazer o mapeamento. Por exemplo, na imagem em cima podemos ter que o titulo é “1.6 MAXILA: VISTA ANTERIOR” só que depois na designação o titulo é “1.6 OSSO MAXILA: VISTA ANTERIOR”.



Ou seja, a palavra “osso” no meio faz com que seja impossível fazer um mapeamento mesmo com funções de uniformização. Isto faz com que no ficheiro JSON resultante da extração deste ficheiro existam entradas que apenas têm as posições sem designação e designações sem descrição.

Uma possível solução para este problema, mas que não chegamos a implementar, seria analisar todos estes casos e criar uma tabela de mapeamento estes casos específicos.

A estrutura de output para este caso é um dicionário contendo o título uniformizado como chave e como valor um dicionário com ids e imagens. Os ids são novamente um dicionário com a letra e respetiva designação e o dicionário de imagens uma lista com o nome e respetiva posição da imagem.


```
{
  "1.1 CRÂNIO: VISTA ANTERIOR": {
    "ids": {
      "a": {
        "descr": "Osso frontal",
        "position": {
          "top": "202",
          "left": "402"
        }
      },
      "b": {
      "c": {
      "d": {
      "e": {
      "f": {
      "g": {
      "h": {
        "descr": "Osso mandíbula",
        "position": {
          "top": "687",
          "left": "427"
        }
      }
    },
    "imgs": [
      {
        "name": "ossos-12_1.jpg",
        "position": {
          "top": "96",
          "left": "204",
          "width": "400",
          "height": "655"
        }
      }
```


7. RU5HW615037.pdf

O documento é um glossário de termos em inglês para português. O processamento deste documento foi o mais fácil de todos. Utilizamos o formato de texto, dado que, é um

padrão muito simples, como por exemplo, “Abdominal Cavity - Cavidade Abdominal” o qual é facilmente extraído por uma expressão regular básica como '(.+) - (.+)'.

Como resultado, geramos um ficheiro JSON com a seguinte estrutura:


```
{
  "abdominal cavity": {
    "pt": "Cavidade Abdominal"
  },
```


8. WIPOPearl_COVID-19_Glossary.pdf

Este documento é novamente um glossário com termos em inglês, em alguns casos seguido de um sinónimo (syn.), seguido de uma descrição, uma tag de classificação, por exemplo “MEDI, Pathology” e por fim várias traduções em diferentes línguas (nove).

Aqui escolhemos novamente o formato XML por ser mais fácil para a identificação do padrão. Após a analise e identificação do padrão para cada bloco de informação, a expressão regular apesar de extensa, como podemos ver na imagem abaixo, não obtivemos grandes problemas na sua extração dos dados neste documento.

```
LANG = ['AR', 'DE', 'ES', 'FR', 'JA', 'KO', 'PT', 'RU', 'ZH']


def process_translations(text):
    term_expr = r'((?:<text [^>]* height="15" font="8"><b>[^<]+</b></text>\s*)+)'
    syn_expr = r'(<text [^>]*><i>\(syn\.\)\s*</i></text>\s*<text [^>]*>[^<]+</text>\s*)?'
    descr_expr = r'((?:<text [^>]* height="12" font="6">[^<]+</text>\s*)+)'
    tag_expr = r'((?:<text [^>]* height="11" font="11">[^<]+</text>\s*)+)'
    not_end_of_file = r'(?!<text top="958")'
    trad_expr = r'((?:' + not_end_of_file + r'[^>]*>(?:<i>)?[^<]+(?:</i>)?</text>\s*)+)'
    all_lang_expr = ""
    for lang in LANG:
        all_lang_expr += r'<text [^>]*><b>\s*(' + lang + r')\s*</b></text>\s*' + trad_expr

    matches = re.findall(term_expr + syn_expr + descr_expr + tag_expr + all_lang_expr, text, re.MULTILINE)
```

Como resultado, é gerado um documento JSON com a seguinte estrutura:


```
{
  "acute respiratory distress syndrome": {
    "syn": "(syn.) ARDS",
    "descr": "Respiratory disease characterized by the rapid onset of widespread inflammation in the lungs.",
    "tag": "MEDI, Pathology",
    "il8n": {
      "ar": "امزلاكمةقنافلايسفنتلاءاحلا",
      "de": "akutes Atemnotsyndrom des Erwachsenen, (syn.) ARDS",
      "es": "síndrome de dificultad respiratoria aguda, (syn.) SDRA",
      "fr": "syndrome de détresse respiratoire aiguë, (syn.) SDRA",
      "ja": "急性呼吸窮迫症候群, (syn.) 急性呼吸促迫症候群, ARDS",
      "ko": "급성  호흡곤란  증후군, (syn.) ARDS",
      "pt": "síndrome do desconforto respiratório agudo, (syn.) SDRA",
      "ru": "острый респираторный дистресс-синдром, (syn.) ОРДС",
      "zh": "急性呼吸窘迫综合征, (syn.) ARDS"
    }
  },
  "ageusia": {
    "syn": "",
    "descr": "Disorder characterized by loss of taste.",
    "tag": "MEDI, Pathology",
    "il8n": {
      "ar": "ةقئاذلا دقفناحقوذلا، دقفناحقوذلا)syn.)",
      "de": "Ageusie",
      "es": "ageusia",
      "fr": "agueusie",
      "ja": "味覚消失",
      "ko": "무미각증",
      "pt": "ageusia",
      "ru": "агевзия",
      "zh": "味觉丧失"
    }
  },
```

3.2. Fase 2: Análise e Correlação da Informação

Após a extração de informações relevantes em cada um dos documentos, foi necessário analisar como relacionar todas as informações para criar uma estrutura de dados. Foram realizadas verificações dos dados comuns e possíveis estruturas de dados para suportar todas as informações e analisar o que seria necessário alterar na estrutura de output de cada um dos ficheiros processados para facilitar a junção das informações.

Após a análise, verificou-se que havia três glossários que poderiam ser combinados, uma vez que são compostos por uma chave em comum, que são termos em Inglês e respetivas traduções em diferentes línguas:

- RU5HW615037 (Inglês - Português)
- CIH Bilingual Medical Glossary English-Spanish (Inglês - Espanhol)
- WIPOPearl_COVID-19_Glossary (Inglês – Português, Espanhol, Francês…)

O resultado final é uma estrutura semelhante ao output do ficheiro WIPOPearl_COVID-19_Glossary, na qual cada termo apresenta as traduções resultantes da junção de todos os ficheiros processados. Para viabilizar esta junção, foi necessário uniformizar todos os termos para letra minúscula e alterar a estrutura de output de cada um deles individualmente, de modo que a fusão fosse possível.


```
"zoonosis": {
  "syn": "",
  "descr": "Infectious disease that is transmissible under natural conditions from vertebrate animals to humans.",
  "tag": "MEDI, Pathology",
  "il8n": {
    "ar": "ضرسينا وبحر دصملا",
    "de": "Zoonose",
    "es": "zoonosis",
    "fr": "zoonose",
    "ja": "人獣共通感染症, (syn.) 動物由来感染症",
    "ko": "인수공통전염병",
    "pt": "Zoonose",
    "ru": "ЗООНОЗ",
    "zh": "人畜共患病"
  }
},
"abdominal cavity": {
"abdominal echotomography": {
"abdominal tuberculosis": {
"abdominal ultrasonography": {
"abdominalgia; celialgia; abdominal pain": {
"abortion": {
  "il8n": {
    "pt": "Aborto",
    "es": "Aborto"
  }
},
"abscess": {
  "il8n": {
    "pt": "Abcesso",
    "es": "Absceso, flemón"
  }
},
```


Temos um quarto glossário “Glossário de Termos Médicos Técnicos e Populares”, no entanto, neste ficheiro o termo está em português, portanto não conseguimos juntar com os restantes glossários que têm o termo em Inglês.

Além disso, identificamos que tínhamos dois seguintes dicionários com o termo em português, onde no primeiro temos traduções e no segundo uma descrição do significado em português.

- dicionario_termos_medicos_pt_en_es
- Dicionario_de_termos_medicos_e_de_enfermagem

Por fim, consideramos que apesar do documento “Glossário de Termos Médicos Técnicos e Populares” ser, como o nome indica, um glossário, poderia ser interessante juntar os termos populares ao dicionário de português.

Como resultado temos a seguinte estrutura:


```
"a, an": {
  "descr": "Prefixo indicando “ausência”. Ex.: amenorréia (falta de menstruação); anoxia (falta de oxigênio)."
},
"aa": {
  "descr": "Abreviatura que os médicos usam nas receitas e que significa “partes iguais”."
},
"abasia": {
  "descr": "Falta de coordenação no andar."
},
"abdome": {
  "descr": "Cavidade oval situada entre o limite inferior do tórax e a pelve. Fica protegido, anterior e lateralmente,
  pelos músculos abdominais e, posteriormente, pelas vértebras e músculos da espinha dorsal. Abriga o estômago, os
  intestinos grosso e delgado, o fígado, a vesícula biliar, o pâncreas, o baço, os rins com as correspondentes
  glândulas supra-renais, a aorta abdominal, vasos sangüíneos e nervos do sistema vegetativo e simpático."
},
"abdome agudo": {
  "descr": "Emergência cirúrgica resultante de distúrbios nas vísceras do abdome."
},
"abdominal": {
  "descr": "Que se refere ou diz respeito ao abdome.",
  "en": "abdominal",
  "es": "abdominal",
  "pop": "ventral"
```


Na restante informação não conseguimos identificar formas de a juntar, por isso passamos para a fase final de criação da estrutura final que contemplasse a informação junta e as informações distintas.

3.3. Fase 3: Definição da estrutura final e armazenamento de dados

Nesta fase, o foco foi na definição da estrutura final do arquivo json que armazenaria os dados extraídos. A estrutura final está representada em baixo:

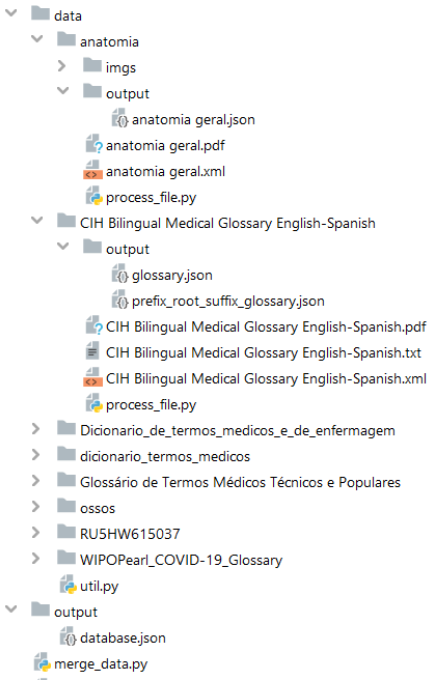

```
{
"Glossary":{...
"Prefixes":{...
"Roots":{...
"Suffixes":{...
"dictionary":{
"pt":{
"en":{...
"es":{...
},
"anatomy": [...
"bones":{...
}
```

Esta estrutura foi definida com base nas informações extraídas dos documentos pdf e na necessidade de organizar os dados de forma clara e consistente para futura utilização. A definição da estrutura foi essencial para o armazenamento dos dados no formato json.

Abaixo encontra-se representada uma porção do output do ficheiro json consolidado.

3. Estrutura do Projeto

Este trabalho prático foi desenvolvido em Python com o uso da biblioteca "re" para a utilização de expressões regulares. Em termos de organização, temos a seguinte estrutura:



- **data** – contém uma diretoria por cada ficheiro analisado com:
    - <diretoria com o nome do ficheiro a processar>
        - *process_file.py* – ficheiro que faz a extração dos dados.
        - **output** – diretoria com o JSON de output gerado.
    - *util.py* – utilitário com funções comuns usadas por todos os processadores.

- **merge_data.py** – ficheiro que faz a junção dos vários outputs na estrutura final.
- **Output**
    - *database.json* – ficheiro que contém a estrutura final e todos os dados extraídos.

4. Conclusão

A realização deste projeto permitiu-nos perceber que a aplicação de expressões regulares no processamento de linguagem natural em documentos biomédicos pode ser uma ferramenta altamente eficaz para a extração de informações relevantes em ficheiros com uma grande quantidade de dados.

Ao longo da extração de dados, encontramos vários desafios devido à complexidade dos dados textuais nos diversos documentos. No entanto, conseguimos alcançar o objetivo de extrair a informação necessária de vários ficheiros PDF e consolidá-la num formato JSON. O formato JSON apresenta enumeras vantagens, como por exemplo a compatibilidade com diferentes linguagens de programação, o que facilita a manipulação dos dados e a sua integração em diferentes sistemas.

Por último, foi possível aplicar os conhecimentos obtidos nas aulas e aprimorar habilidades de análise de dados e programação.