Universidade do Minho
Escola de Engenharia

Trabalho Prático

Processamento de Linguagem Natural em Engenharia Biomédica

Membros do grupo:

Ana Beatiz Salgado Andrade PG56107

Filipa José Rodrigues de Araújo Costa PG56123

Leonor de Amorim Pereira PG57813

Professores:

Luís Filipe Cunha

José João Almeida

Conteúdo

1 Introdução

O presente relatório descreve o desenvolvimento do Trabalho Prático 1 da unidade curricular de Processamento de Linguagem Natural em Engenharia Biomédica.

Este projeto teve como principal objetivo a aplicação de técnicas de processamento de linguagem natural para extrair informação relevante de documentos em formato PDF, com foco na área biomédica.

Ao longo do trabalho, foram construídas ferramentas capazes de analisar e transformar documentos complexos em estruturas de dados organizadas, armazenadas no formato JSON. Este processo envolveu a análise detalhada dos documentos, a definição de uma estrutura de dados adequada, a conversão e limpeza do conteúdo textual, e a extração sistemática de informação. O resultado final visa não só demonstrar a compreensão das técnicas abordadas nas aulas, mas também fornecer uma base reutilizável para trabalhos futuros.

A implementação foi realizada em linguagem *Python*, respeitando as restrições definidas no enunciado, e incluiu o processamento obrigatório dos ficheiros *diccionari-multilinguee-de-la-covid-19.pdf* e *glossario_neologismos_saude.pdf*.

2 *diccionari-multilinguee-de-la-covid-19.pdf*

Foi realizado um processo de extração de informação estruturada a partir de um documento em formato PDF. Este trabalho foi dividido em várias etapas, desde a limpeza do conteúdo até à conversão para uma estrutura hierárquica em JSON, com o apoio de *scripts* desenvolvidos em *Python.*

2.1 Seleção e Estrutura da Informação Relevante

Inicialmente, procedeu-se à análise da estrutura do dicionário original, de forma a compreender o padrão dos dados a extrair. Cada entrada do dicionário segue uma organização específica:

- Cada palavra começa por um número identificador (número do conceito).
- Segue-se a palavra em catalão (a *bold*), acompanhada da respetiva categoria gramatical (em itálico).
- Sinónimos e outras variantes (sinónimos complementares, absolutos ou denominação comercial, siglas ou até consulta de outras entradas) aparecem numa linha distinta, cada um com a sua categoria lexical.
- As traduções aparecem a seguir, uma por linha, com: língua (em itálico), tradução e categoria lexical (traduções em português de Portugal e português do Brasil são diferenciadas com [PT] e [BR]).
- De seguida, pode existir uma secção de códigos alternativos (símbolos, nome científico ou até código CAS).
- A definição surge a seguir, começando por uma área temática (ex.: Clínica, Diagnóstico, etc.), seguida da descrição do conceito.
- Por fim, podem aparecer notas adicionais, assinaladas com o prefixo "Nota:".

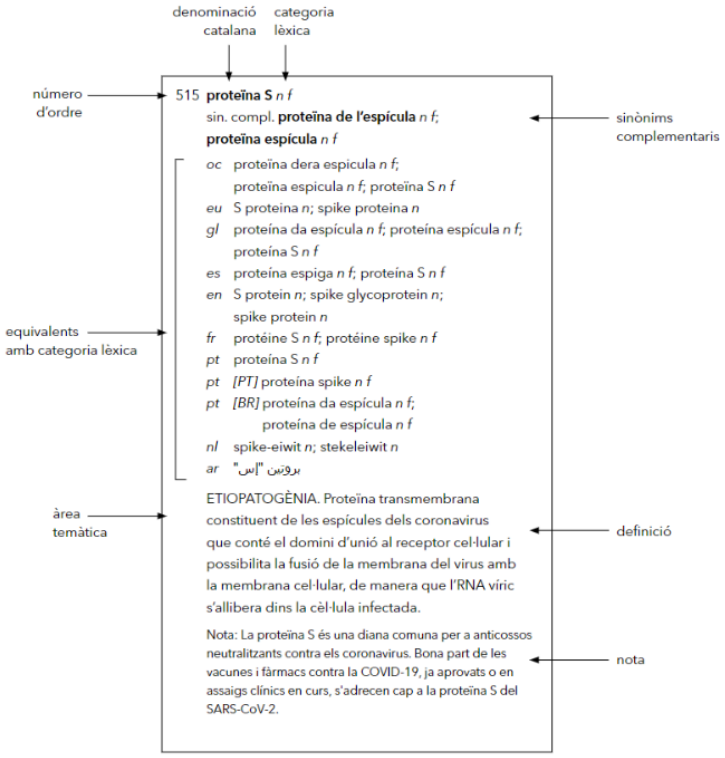

Figura 1: Estrutura geral dos conceitos no dicionário

2.2 Definição da Estrutura de Dados

Após a análise estrutural, definiu-se uma sintaxe JSON para representar a informação extraída, *bd_struct.json*. Esta estrutura organiza os dados por letra inicial e número de conceito, com campos como designação, categoria lexical, traduções, códigos alternativos, áreas temáticas e notas.

Exemplo resumido:

```
{"a": {
        "1": {
        ...
        },
        "2": {
            "designacao": "acalabrutinib",
            "categoria lexical da designacao": "n m (nom masculi)",
            "complementos designacao": {
                "siglas": [],
                "sinonimos absolutos": "",
                "sinonimos complementares": [],
```

```
            "denominacao comercial": "",
            "consultar outra entrada": ""
        },
        "traducoes": {
            "occita": "acalabrutinib n m",
            "basc": "akalabrutinib n",
            "gallec": "acalabrutinib n m",
            "castella": "acalabrutinib n m",
            "angles": "acalabrutinib n m",
            "frances": "acalabrutinib n m",
            "[PT]portugues de Portugal": "acalabrutinib n m",
            "[BR]portugues do Brasil": "acalabrutinibe n m",
            "neerlandes": "acalabrutinib n",
            "arab": ""
        },
        "codigos alternativos": {
            "simbolo": "",
            "nome cientifico": "",
            "numero CAS": "1420477-60-6"
        },
        "areas tematicas": {
            "area": "Principis Actius",
            "descricao": "Farmac antineoplastic que bloca\nla
...."
        },
        "notas": [
            "1.  L  acalabrutinib  s empra  en el tractament\nde ..."
        ]
    },
    "3":{
    ...
    },
    ...}
```

2.3 Conversão do PDF para Texto

Utilizou-se um *script* específico, presente no ficheiro *pdftotext.py*, para converter o ficheiro PDF original em texto plano. Esta etapa foi essencial para que o conteúdo pudesse ser posteriormente processado linha a linha, com o objetivo de extrair e estruturar a informação de forma automatizada.

O *script* foi desenvolvido com base na biblioteca *PyMuPDF (fitz)* e permite extrair texto de documentos com duas colunas, mantendo a ordem lógica de leitura, primeiro a coluna da esquerda, depois a da direita e, por fim, a passagem para a página seguinte. O processamento ignora as duas primeiras páginas físicas (geralmente capa e prefácio), considerando que a numeração lógica da primeira página relevante começa a partir da terceira página física.

Cada página é dividida em duas metades com base na largura total, e o conteúdo é separado em colunas "esquerda"e "direita"de acordo com a posição horizontal dos blocos de texto. Os blocos são classificados com base na sua posição vertical (*bbox*) para garantir que a ordem de leitura original seja respeitada.

Além disso, o *script* adiciona um cabeçalho com o número lógico da página (ex.: === PÁGINA 1 ===) antes de inserir o conteúdo textual, o que permitirá, posteriormente, eliminar toda a informação irrelevante mais facilmente, especificado melhor numa secção mais à frente. Cada coluna é processada individualmente, com as suas linhas ordenadas verticalmente e posteriormente combinadas. O texto final resultante é guardado num ficheiro *.txt* com codificação UTF-8, preservando a legibilidade e a estrutura visual do dicionário original.

2.4 Limpeza e Preparação dos Dados

Com o texto bruto extraído, foram aplicadas várias operações no script *extracao.py* com o objetivo de limpar e estruturar a informação de forma adequada. Numa primeira fase, removeram-se caracteres de controlo herdados de sistemas antigos, nomeadamente o Form Feed (FF, representado por \x0C) e o Bell (BEL, representado por \x07), recorrendo a expressões regulares, Figura 2.

Posteriormente, eliminou-se toda a informação irrelevante que se encontrava antes do marcador "=== PÁGINA 28 ===", ponto a partir do qual se inicia a parte útil do dicionário, bem como tudo o que surgia após o marcador "=== PÁGINA 181 ===", Figura 3. Estes marcadores foram inseridos durante a conversão do PDF para texto, facilitando assim a delimitação do conteúdo relevante.

Adicionalmente, foram removidas as numerações de páginas que surgiam sobrepostas às marcas automáticas geradas na extração, Figura 4, bem como os títulos repetitivos de cabeçalho, como "QUADERNS 50 DICCIONARI MULTILINGÜE DE LA COVID-19", que apareciam no início de cada nova página.

Alguns ajustes manuais também foram feitos de forma a facilitar a extração da informação relevante (agrupamento de número e designação na mesma linha nos primeiros 100 conceitos (Figura 5), remoção de casos anómalos (onde a numeração original está presente no meio dos conceitos - Figura 6) e tratamento de hífens (Figura 7)).

Após este processo de limpeza, foi utilizada uma expressão regular para extrair os conceitos do dicionário de forma estruturada, identificando três componentes principais: o número do conceito, a designação e a categoria gramatical (como substantivo masculino ou feminino, adjetivo, verbo transitivo, intransitivo, entre outros). Por fim, o texto limpo foi guardado num novo ficheiro, *dicionario_limpo.txt*, ficando assim pronto para ser processado automaticamente em fases posteriores.

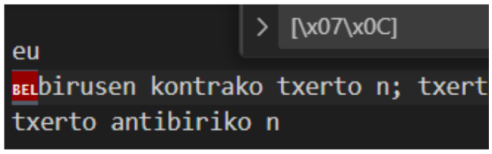

Figura 2: Formatação herdada na conversão de pdf para texto

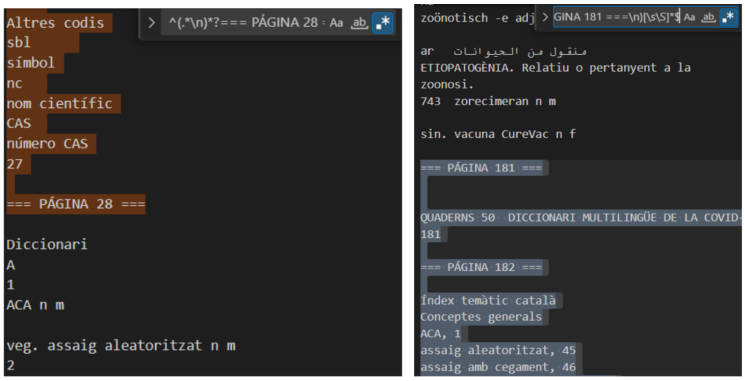

Figura 3: Informação irrelevante extraída

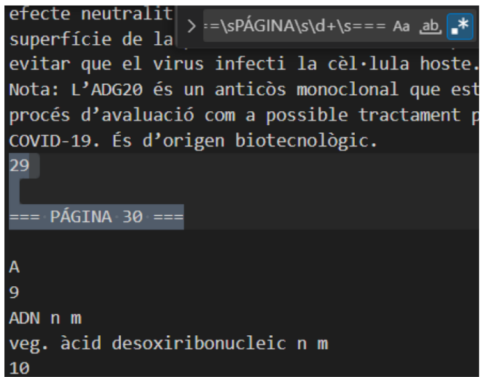

Figura 4: Paginação incial juntamente com a paginação formulada na extração

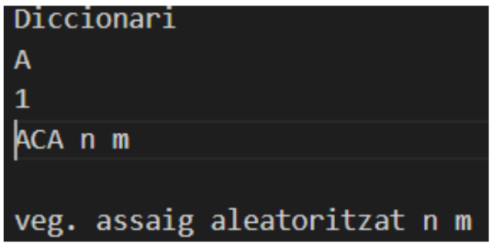

Figura 5: Primeiros 100 conceitos encontram-se com a numeração e designação em linhas distintas

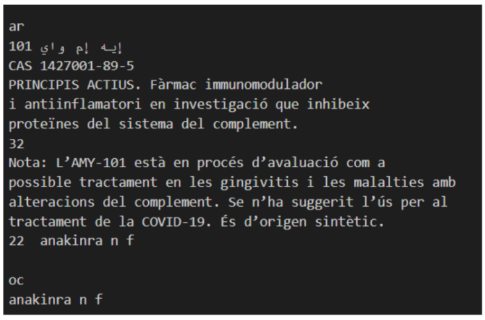

Figura 6: Paginação original extraida do documento

Figura 7: Casos especiais de caracteres distintos, como hífen entre outros

2.5 Extração e Estruturação com Python

Após a conversão do dicionário de pdf para texto e uma limpeza inicial, utilizou-se um script desenvolvido em *Python* para estruturar os dados num formato hierarquizado e interpretável. O processamento teve por base a biblioteca *defaultdict* da coleção collections, permitindo organizar os dados por letra inicial e número do conceito. A análise foi realizada linha a linha, com extensivo uso de expressões regulares (*regex*) para identificar diferentes elementos linguísticos e metainformação.

Cada entrada do dicionário foi classificada com base em padrões específicos, que permitiram distinguir entre designação do conceito, categoria lexical (ex. "n f", "adj", "v tr/intr", etc), traduções multilingues, sinónimos absolutos e complementares, siglas, áreas temáticas, notas explicativas, referências cruzadas ("veg."), e códigos internacionais como o CAS (*Chemical Abstracts Service*). O código inclui um mapeamento de abreviações lexicais para descrições completas em português, o que facilita a compreensão e posterior utilização dos dados.

As traduções foram processadas com especial atenção: ao detetar a abreviatura da lín-

gua (como fr, en, es, ar, ou variantes como [PT] e [BR]), o *script* ativa uma fase de acumulação de linhas, armazenando-as num *buffer*. Funções auxiliares específicas, como *finalizar_traducao_arabe()* e *finalizar_outra_traducao()*, garantem que cada bloco de tradução é corretamente encerrado e armazenado no dicionário antes do início de um novo conceito ou secção. A função *finalizar_traducoes_pendentes()* é chamada de forma preventiva sempre que há transição entre letras, conceitos ou secções, evitando a perda de informação parcial.

A função principal, *processar_documento()*, controla todo o fluxo de *parsing*, reconhecendo secções com *nova_secao()*, e atribuindo corretamente os conteúdos extraídos às respetivas chaves dentro da estrutura final.

O dicionário é finalmente guardado num ficheiro *.json*, permitindo reutilização em análises, visualizações ou aplicações *web*.

3 m_glossario-tematico-monitoramento-e-avaliacao.pdf

Após a análise da organização interna do glossário original, foi definida uma estrutura uniforme para representação das entradas, com base nos elementos semânticos e formais presentes. Cada entrada segue o seguinte modelo de extração e codificação:

- **Entrada (chave principal do glossário)**

  Representa a palavra que está a ser definida e descrita no glossário. Corresponde à unidade linguística central de cada entrada, o termo principal da linguagem de especialidade que contém o conteúdo semântico a ser explicado.

- **Género (`tipo`)**

  Indica o género gramatical do termo da língua descrita: masculino (`masc`), feminino (`fem`) ou ambos (`fem./masc.`). Esta informação é extraída da notação gramatical associada ao termo.

- **Número (`número gramatical`)**

  Indica se o termo é usado exclusivamente no plural, sendo assinalado pela marcação `pl`. Quando presente, o campo assume o valor "plural".

- **Descrição (`descrição`)**

  Corresponde à definição do termo, descrevendo os traços essenciais do conceito no seu domínio técnico ou temático. Pode incluir explicações contextuais, funcionais ou classificatórias.

- **Sinónimo (`singular`)**

  Assinala a presença de marcações como `Sin.`, `Sin .`, `Ver sin.` ou `Ver sin .`, que indicam uma relação de sinonímia com outro termo. Este campo armazena apenas a marcação, e não o termo sinónimo em si, o conteúdo seguinte à marca integra-se na `descrição`.

- **Remissiva (`remissiva`)**

  Indica uma relação de remissão geral com outro termo, assinalada por "Ver". O campo contém apenas o termo de destino, para o qual o leitor deve ser direcionado.

- **Definição Expandida**

  Identificada por uma seta, esta marcação indica que a entrada remete para a forma expandida de uma sigla ou abreviatura. Quando presente, o campo descrição permanece vazio.

- **Nota (`notas`)**

  Apresenta informação adicional ao conceito, podendo ter natureza linguística, técnica, prática ou enciclopédica. É precedida pelas marcações `Nota:` ou `Notas:`.

- **Equivalentes em Língua Estrangeira (`espanhol`, `inglês`)**

  Regista os termos em espanhol e inglês considerados conceitualmente equivalentes ao termo em português. São identificados pelos marcadores `Em espanhol:` e `Em inglês:`.

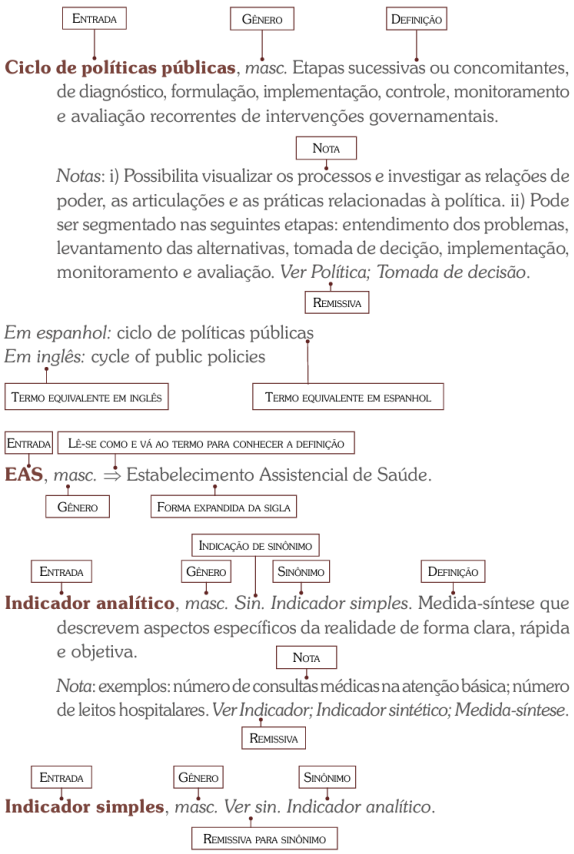

Figura 8: Estrutura geral dos Conceitos no Dicionário.

3.1 Definição da Estrutura de Dados

Após a extração e organização da informação lexical a partir do glossário original, definiu-se uma estrutura sintática em formato JSON para armazenar os dados de forma hierárquica e pesquisável. A base de dados resultante, designada ***glossario.json***, está organizada por letra inicial da entrada (termo principal), sendo que cada uma contém as respetivas palavras associadas a essa letra.

Cada entrada é representada como um objeto contendo os seguintes campos:

- "tipo": indica o género gramatical do termo (ex: "masc", "fem").
- "número": especifica se o termo é plural (ex: "plural").
- "descrição": descrição textual do conceito ou termo.
- "sinônimo": marcações de sinonímia presentes na entrada (ex: "Sin.", "Ver sin."), mas não o termo sinónimo em si.
- "remissiva": termo associado a uma remissão por meio da marcação "Ver".

- "expandida": forma expandida do termo, identificada pela marca de uma seta.
- "notas": informações adicionais sobre o termo (comentários práticos, linguísticos ou enciclopédicos).
- "espanhol": equivalente em espanhol do termo.
- "inglês": equivalente em inglês do termo.

Exemplo Resumido:


```
{
  "A": {
    "Acao": {
      "tipo": "fem",
      "numero": "",
      "descricao": "Opera  o que resulta em produto voltado para..",
      "notas": "i) Pode ser um projeto, uma atividade, uma ...",
      "espanhol": "accion",
      "ingles": "action",
      "sinonimo": "",
      "remissiva": "Atividades; Programa .",
      "expandida": ""
    },
    "Alocacao de recursos": {
      "tipo": "fem",
      "numero": "",
      "descricao": "Destina  o de recursos. ...",
      "notas": "i) Termo muito utilizado nos campos da economia ...",
      "espanhol": "asignaci n de recursos",
      "ingles": "allocation of resources",
      "sinonimo": "Sin.",
      "remissiva": "",
      "expandida": ""
    }
    ...
  },
```

```
"D": {
  "Destinacao de recursos": {
    "tipo": "fem",
    "numero": "",
    "descricao": "Alocacao de recursos.",
    "notas": "",
    "espanhol": "",
    "ingles": "",
    "sinonimo": "sin.",
    "remissiva": "",
    "expandida": ""
  }
  ...
},
"E": {
  "EAS": {
    "tipo": "masc",
    "numero": "",
    "descricao": "",
    "notas": "",
    "espanhol": "",
    "ingles": "",
    "sinonimo": "",
    "remissiva": "",
    "expandida": "Estabelecimento Assistencial de Saude."
  },
  "Entregas intermediarias": {
    "tipo": "fem",
    "numero": "plural",
    "descricao": "Resultados intermediarios.",
    "notas": "",
    "espanhol": "",
    "ingles": "",
```

```
        "sinonimo": "sin.",
        "remissiva": "",
        "expandida": ""
      }
    },
    ...
}.
```

3.2 Conversão do PDF para Texto

A conversão do ficheiro PDF para texto plano foi realizada através de um *script* desenvolvido com a biblioteca *PyMuPDF (fitz)*. O objetivo desta etapa era extrair o conteúdo textual de um documento estruturado, mantendo a lógica de leitura e organização do dicionário original. Para isso, foi utilizado um intervalo específico de páginas — da 21 à 79 — que correspondem ao conteúdo relevante do glossário, excluindo capas, prefácio e outras partes não essenciais.

Cada página foi analisada como um conjunto de blocos de texto, a partir dos quais se extraiu informação linha a linha e palavra a palavra. Foram utilizadas informações como a posição dos blocos e a cor da fonte para determinar quais elementos correspondiam a palavras-chave e quais faziam parte das definições. Uma lógica de rastreamento foi aplicada para manter a integridade da estrutura original, garantindo que cada entrada no dicionário fosse corretamente identificada e agrupada.

Além disso, foi implementado um sistema de organização alfabética, no qual cada nova palavra iniciada por uma letra diferente introduz um cabeçalho correspondente (ex.: A, B, C, etc.). Este detalhe foi fundamental para preservar a navegabilidade do dicionário e facilitar futuras consultas ao conteúdo extraído. Ao final do processo, o resultado foi gravado num ficheiro *.txt* com codificação UTF-8, pronto para ser utilizado noutras etapas do projeto.

3.3 Limpeza e Preparação dos Dados

A etapa de limpeza e preparação dos dados esteve integrada no próprio processo de extração textual, com o objetivo de eliminar ruído e garantir que apenas a informação relevante fosse considerada na construção do dicionário. Foram aplicadas diversas regras de filtragem baseadas em características do conteúdo, como o comprimento do texto, a posição na página e a cor do elemento.

Elementos considerados ruído, como a paginação, marcadores de secção e palavras incompletas ou deslocadas ("onitoramento", "valiação", "lossário", "emático"), foram explicitamente excluídos. Também foi descartado qualquer conteúdo cuja cor correspondesse ao branco (RGB 16777215), maior parte desse conteúdo estava em marcadores de secção o que torna essa informação irrelevante.

Outro ponto crítico foi a diferenciação entre os títulos dos verbetes e os seus significados. Para isso, utilizou-se a cor da fonte como critério: textos com uma cor específica (8470328) foram interpretados como palavras-chave, enquanto os restantes constituíam os seus significados. Quando uma nova palavra era identificada, a definição da anterior era finalizada e gravada no ficheiro, iniciando-se uma nova entrada.

Importa ainda destacar que apenas as páginas entre a 21.ª e a 79.ª foram consideradas no processamento, tendo sido deliberadamente ignoradas as páginas anteriores e posteriores a esse intervalo, por conterem informações como capa, sumário, prefácio, apêndices ou anexos, que não faziam parte do glossário em si. Este controlo contribuiu significativamente para a precisão e qualidade do resultado final.

Como resultado, o texto limpo foi guardado no ficheiro, *dicionario_formatado.txt*, ficando assim pronto para ser processado automaticamente em fases posteriores.

3.4 Extração e Estruturação com *Python*

Com o dicionário já extraído em formato *.txt*, avançou-se para a fase de estruturação e normalização da informação, com o objetivo de transformar os dados textuais num formato estruturado, neste caso, JSON.

Recorreu-se à linguagem *Python* para desenvolver um script que percorre, linha a linha, o conteúdo do ficheiro de entrada, identificando os diferentes componentes de cada entrada do glossário através de expressões regulares cuidadosamente definidas. Esta abordagem permitiu extrair, de forma semi-automatizada, os seguintes campos: palavra principal, tipo gramatical, número (singular ou plural), descrição, notas adicionais, equivalentes em espanhol e inglês, sinónimos, remissivas (entradas que remetem para outras) e definições expandidas.

A lógica de deteção baseou-se, principalmente, em padrões textuais recorrentes observados no glossário original, como o uso de marcadores fixos ("Notas:", "Em espanhol:", "Em inglês:", "Sin.", "Ver sin.", entre outros) para segmentar e categorizar a informação. Sempre que identificados, esses marcadores serviram como delimitadores para extrair os respetivos

conteúdos. A organização alfabética foi também mantida, sendo cada letra associada ao seu respetivo subconjunto de palavras.

Este processo culminou na criação de um dicionário *Python*, que foi posteriormente exportado para um ficheiro *glossario.json*, com estrutura hierárquica e indentação legível. O resultado final permite um acesso mais eficiente e programático aos dados do glossário.

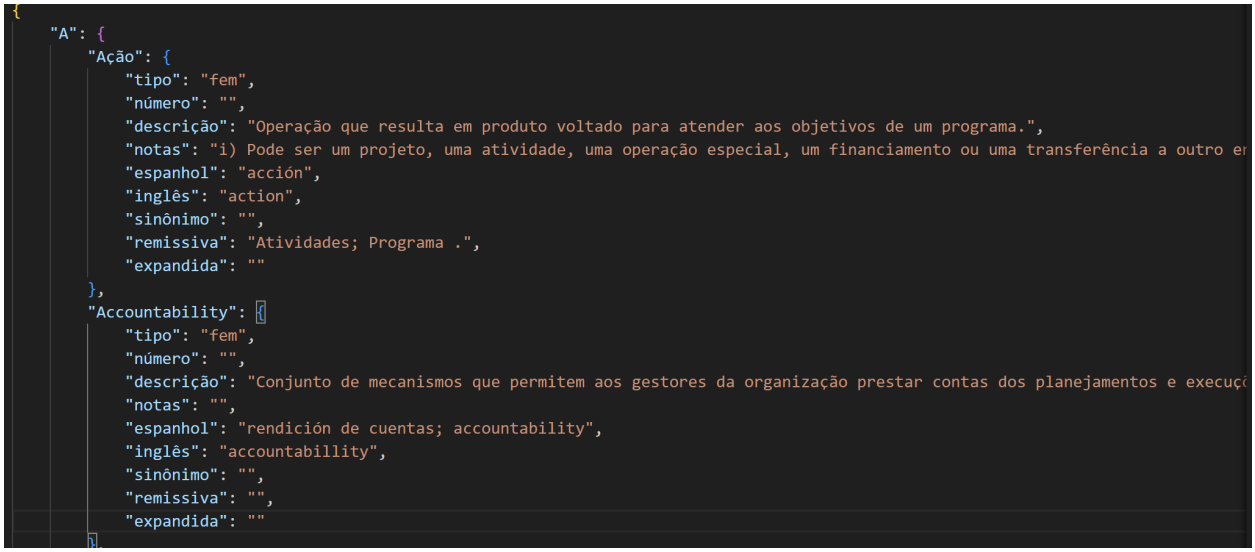

Figura 9: Excerto do "Glossario.json".

4 glossario_neologismos_saude.pdf

4.1 Definição e Estrutura dos Dados

A estruturação dos dados do *Glossário de Neologismos Terminológicos da Saúde Humana* (GNTSH) está organizada em múltiplos campos. Para garantir que nenhuma informação fosse perdida durante o processo de extração, foi adotado um modelo de dados que mantém os principais elementos definidos na organização original do glossário, assegurando a consistência, acessibilidade e a possibilidade de reutilização.

Cada entrada foi representada como um objeto com campos específicos, estruturados segundo a seguinte lógica:

```
{
    "entrada": {
        "referencia_gramatical": "",
        "equivalencias": {
            "ingles": "",
            "espanhol": ""
        },
        "Sigla": "",
        "Definicao": "",
        "Informacao_enciclopedica": "",
        "Abonacao": "",
        "Sinonimos": "",
        "Numero_identificacao": "",
        "Marcas Tipográficas": ""
    }
}
```

Esta estrutura reflete os elementos definidos no glossário:

- **entrada**: representa o lema, geralmente um substantivo, apresentado em letras minúsculas, que pode ser simples ou composto.
- **referência gramatical**: logo após a entrada, indica o gênero do substantivo (*s.m.* para masculino e *s.f.* para feminino).

- **equivalências**: correspondem às traduções do termo em inglês (`[ing]`) e espanhol (`[esp]`), aparecendo em itálico no glossário.
- **sigla**: campo opcional, aplicado em termos frequentemente representados por acrónimos no discurso técnico.
- **definição**: apresenta a explicação do termo com base num hiperónimo e nas características específicas do conceito, com linguagem simples e acessível.
- **informação enciclopédica**: campo opcional que complementa a definição com dados contextuais, históricos ou técnicos sobre o termo, inicia normalmente com "Inf. enciclopédica".
- **abonação**: citação retirada da *Revista Pesquisa* que ilustra o uso real do termo. É obrigatório e oferece apoio contextual para a compreensão, encontra-se entre aspas.
- **sinônimos**: campo opcional onde se registam termos equivalentes, com remissões entre eles quando apropriado. Inicia-se com "Ver este termo".
- **número de identificação**: remete ao número da matéria da *Revista Pesquisa* na qual o termo foi encontrado, sendo um campo obrigatório e indicado entre parênteses.
- **marcas tipográficas**: inclui o tratamento tipográfico dado aos termos na definição, itálico, fundamental para preservar a função linguística e informativa destes recursos visuais.

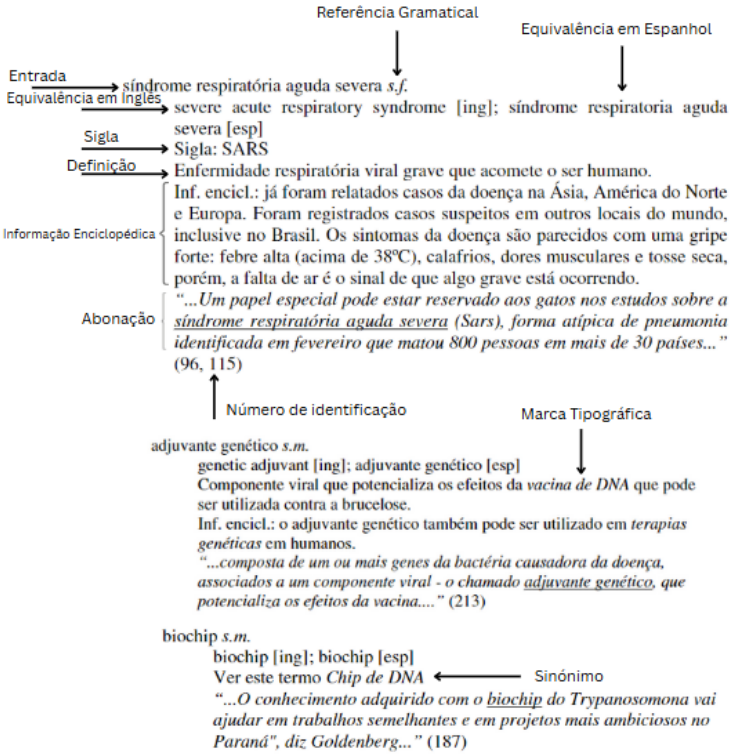

Figura 10: Estrutura geral dos conceitos no glossário.

4.2 Limpeza e Preparação dos Dados

A preparação dos dados teve início com a conversão do glossário original em PDF para um formato de texto simples, utilizando o comando *pdftotext*. Esta conversão permitiu obter uma versão acessível dos conceitos, facilitando a manipulação dos dados. Antes de realizar qualquer tipo de extração automática, foi feita uma limpeza manual criteriosa para remover todo o conteúdo que não fazia parte da secção *"O glossário"*. Esta etapa foi essencial para isolar exclusivamente os conceitos e garantir que apenas as informações relevantes fossem processadas.

Com o conteúdo reduzido aos conceitos, iniciou-se as primeiras extrações estruturadas, baseadas nas regularidades do texto plano. Foram identificados elementos como entrada, referência gramatical, equivalências, definições, abonações, entre outros campos previstos na organização do glossário. Posteriormente, percebeu-se que o documento original continha marcas tipográficas, termos destacados em itálico, que eram relevantes para a representação fiel dos dados, especialmente em definições. Como o arquivo *.txt* gerado pelo pdftotext não preservava essas informações, foi necessário utilizar o comando *pdftohtml*, convertendo o PDF para HTML, o que possibilitou o acesso à formatação original.

A partir do HTML, iniciou-se um processo de limpeza automática: utilizando expressões regulares, foram removidas todas as tags *<text ...>* e *</text>*, mantendo apenas o conteúdo

necessário. Com o HTML limpo, um script percorreu o documento linha a linha para identificar expressões em itálico (marcadas por *...*) e associá-las corretamente às entradas correspondentes.

Durante esta fase, também foram aplicadas normalizações adicionais, como a remoção de espaços em excesso, filtragem de caracteres e entidades HTML irrelevantes, e padronização de formatos. O resultado final desta preparação foi a criação de um arquivo JSON contendo as marcas tipográficas organizadas por entrada, pronto para ser integrado à estrutura geral dos dados extraídos.

4.3 Extração e Estruturação com Python

Com os dados do glossário já disponíveis em formato *.txt* e *.html*, desenvolveu-se um conjunto de *scripts* em *Python* com o objetivo de automatizar a extração e estruturação da informação, convertendo os dados textuais dispersos num formato estruturado e reutilizável, nomeadamente, o formato JSON.

A abordagem adotada recorreu intensivamente ao uso de expressões regulares para identificar os diferentes componentes de cada entrada do glossário. No caso do ficheiro de texto (*glossario.txt*), o script percorre o conteúdo de forma sequencial, identificando entradas através de padrões como *s.f.* ou *s.m.*, que indicam a classe gramatical (substantivo feminino ou masculino). A partir daí, o conteúdo é particionado e analisado consoante a sua função, permitindo extrair, de forma semi-automatizada, a informação.

Cada uma destas componentes é extraída com base em marcadores específicos e regularidades formais detetadas no corpus original. Foram utilizados também mecanismos de normalização, como a eliminação de espaços redundantes, formatação de caracteres especiais e remoção de ruído (ex: aspas não informativas, pontos irrelevantes , etc.). Paralelamente, outro módulo do código opera sobre o ficheiro HTML (*glossario.html*), com o objetivo de isolar marcas tipográficas, expressões em itálico que representam classificações semânticas ou metalinguísticas. Após a remoção de tags indesejadas (*<text>*), o *script* percorre cada entrada e recolhe todas as ocorrências de texto entre *...*, filtrando os conteúdos por tamanho, relevância e formato. As marcas são posteriormente associadas à entrada lexical correspondente.

A etapa final do processo envolve a junção dos dois conjuntos de dados (informação textual e marcas tipográficas), resultando num dicionário Python estruturado, posteriormente exportado como *glossario_completo.json*, com indentação legível e organização alfabética.

Esta estrutura hierárquica permite aceder de forma rápida a qualquer informação relevante do glossário.

4.4 Problemas Encontrados na Exploração do Código

Durante o desenvolvimento do processo de extração e estruturação dos dados, surgiram diversos desafios que exigiram ajustes sucessivos no código para garantir a correta interpretação das entradas. Estes problemas revelam tanto a complexidade do glossário original como a necessidade de um tratamento mais refinado dos dados textuais. Abaixo listam-se os principais obstáculos encontrados:

- **Equivalências distribuídas em múltiplas linhas:** Inicialmente, o código procurava as traduções (equivalentes em inglês e espanhol) apenas na mesma linha da definição, o que resultava em falhas de deteção quando essas equivalências se estendiam por mais de uma linha.

- **Variações na formatação entre conceitos:** As inconsistências na formatação das entradas (uso de maiúsculas, pontuação, espaçamento, etc.) dificultaram a identificação precisa de onde começava e terminava cada campo (definição, tradução, abonação, etc.).

- **Informação enciclopédica e abonação na mesma linha:** Em vários casos, partes com informações enciclopédicas estavam misturadas com a abonação, o que confundia a lógica de separação e exigiu regras específicas para tratar essas sobreposições.

- **Números identificadores com espaços:** Algumas entradas apresentavam números de identificação com espaços intermediários, fazendo com que fossem interpretados incorretamente e que partes importantes da abonação fossem descartadas.

- **Pontuação antes de `[esp]`:** Em alguns casos, existia um ponto final logo antes da marca de equivalência em espanhol, como em "acidose metilmalônica." Isto partia a expressão regular padrão e exigiu adaptação para captar corretamente o conteúdo após o ponto.

- **Aspas irregulares na abonação:** Algumas citações começavam com aspas curvas (ex.: "“") mas terminavam com apóstrofo (ex.: "’"), o que criava dificuldades na delimitação correta da abonação.

- **Aspas dentro das aspas de abonação:** Em entradas como *"biochip"* ou *"biofeedback"*, havia aspas internas na citação, o que confundia o delimitador padrão e exigiu um tratamento especial para preservar o conteúdo sem truncamento.
- **Inconsistência nos rótulos de informação enciclopédica:** Foram encontradas variações como "Inf. Encl.:", "Inf. enciclo.:" e até erros como "Inf. ecic.:", obrigando à criação de padrões flexíveis de reconhecimento, como em *"bipsoro eletrónico"*.
- **Uso de `[es` em vez de `[esp]`:** Algumas traduções estavam marcadas incorretamente com `[es`, exigindo normalização manual ou ajustes de regex para detetar essas variações, como em *"câncer gástrico"*.
- **Diversidade de tipos de aspas na abonação:** Diferentes estilos de aspas iniciais e finais usados ao longo do glossário dificultaram a identificação uniforme dos limites das citações.
- **Aspas dentro da definição:** Em casos como *"clonagem reprodutiva"*, a presença de aspas dentro da própria definição interferia nas delimitações internas do texto.
- **Uso do caracter especial "..." (reticências tipográficas):** A presença do caracter Unicode para reticências (...) nas abonações, como em *"dermatoscopia"*, criava incompatibilidades com os padrões esperados, o que levou à necessidade de normalizar este caracter.
- **Casos Especiais:** Houve alguns casos em que os erros verificados, por se tratarem de casos excecionais, foram corrigidos manualmente, como por exemplo no caso de "encefalopatia espongiforme felina"que a tradução em inglês ficou abaixo, na linha após o `[esp]`.

Estes problemas foram gradualmente solucionados com ajustes no código (ou manualmente, como no caso descrito acima), principalmente por meio do refinamento das expressões regulares e pelo tratamento de exceções. Sendo o resultado final, o *glossario_completo.json*, uma forma eficaz e confiável de ter acesso aos dados do glossário.

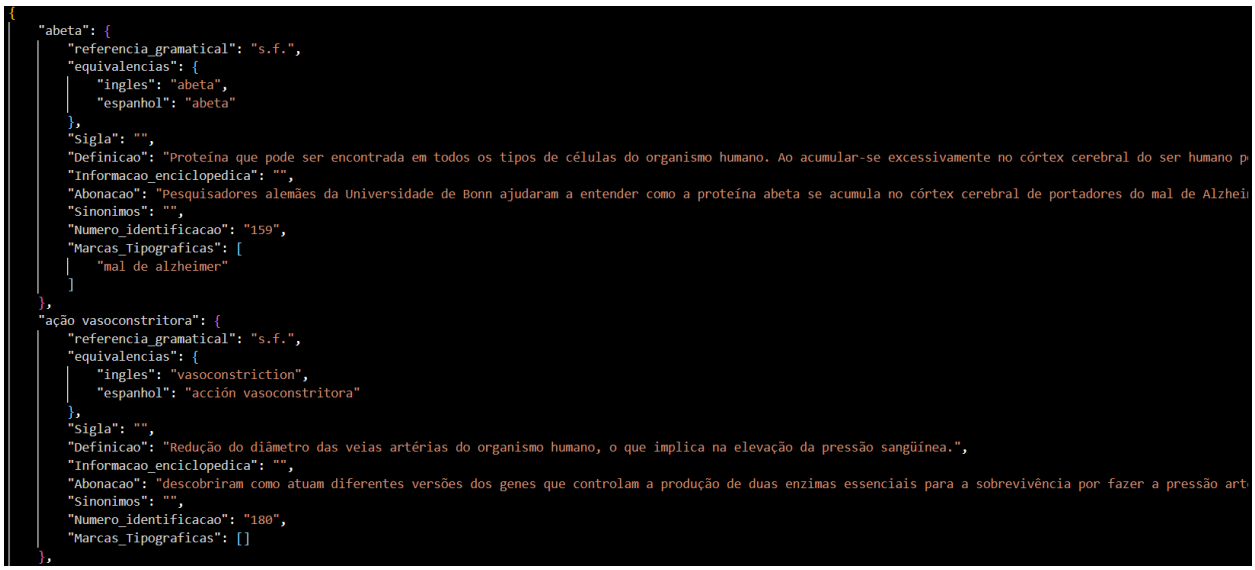

Figura 11: Excerto do "glossario_completo.json".

5 Conclusão

O desenvolvimento deste trabalho prático permitiu aplicar de forma concreta diversos conceitos e técnicas de Processamento de Linguagem Natural, com foco na área biomédica. Através da análise, extração e estruturação de dados oriundos de documentos complexos em formato PDF, foi possível transformar informação não estruturada em recursos digitais organizados e reutilizáveis, com especial destaque para a construção de ficheiros JSON semântica e estruturalmente ricos.

O projeto envolveu diferentes desafios técnicos, como a conversão fiel dos documentos, a eliminação de ruído textual, o reconhecimento de padrões linguísticos e a preservação de marcas tipográficas relevantes. A utilização da linguagem *Python*, aliada ao uso criterioso de expressões regulares e bibliotecas especializadas como *PyMuPDF*, demonstrou-se essencial para o sucesso das tarefas.

Além de consolidar conhecimentos teóricos discutidos em aula, o trabalho proporcionou uma experiência prática relevante em engenharia de dados textuais, reforçando a importância da atenção ao detalhe na extração de conhecimento a partir de fontes heterogéneas.

Consideramos que cumprimos os objetivos propostos e fomos capazes de desenvolver estruturas de dados apropriadas para representar de forma fiel e completa toda a informação presente nos glossários analisados. Os resultados obtidos refletem um esforço contínuo de análise crítica, adaptação e validação das abordagens utilizadas ao longo do processo.