UNIVERSIDADE D
COIMBRA

José Henrique Gomes da Silva Dias Pereira

Extração de Informação em Documentos Não Estruturados

Dissertação no âmbito do Mestrado em Engenharia e Ciência de Dados, orientada pela Professora Catarina Silva, pelo Professor Hugo Amaro e pelo Professor Hugo Oliveira e apresentada ao Departamento de Engenharia Informática da Faculdade de Ciências e Tecnologia da Universidade de Coimbra.

Setembro de 2022

Faculdade de Ciências e Tecnologia

Departamento de Engenharia Informática

Extração de Informação em Documentos Não Estruturados

José Henrique Gomes da Silva Dias Pereira

Dissertação no âmbito do Mestrado em Engenharia e Ciência de Dados, orientada pela Professora Catarina Silva, pelo Professor Hugo Amaro e pelo Professor Hugo Oliveira e apresentada ao Departamento de Engenharia Informática da Faculdade de Ciências e Tecnologia da Universidade de Coimbra.

Setembro 2022

UNIVERSIDADE D
COIMBRA

Resumo

A par da evolução tecnológica dos últimos anos, cada vez mais empresas e indústrias caminham no sentido da automatização dos seus processos, quer seja motivado pela redução dos custos, acréscimo de eficiência ou rapidez. O mesmo acontece em tarefas do domínio linguístico onde avanços recentes cada vez mais aproximam a capacidade de compreensão de um modelo de inteligência artificial à do ser humano. Assim, aproveitando a cada vez maior quantidade de informação disponível online, este trabalho foca-se na extração de informação automática a partir de documentos não estruturados, utilizada na elaboração de relatórios toxicológicos de substâncias químicas. Através de técnicas como reconhecimento de entidades, similaridade semântica, identificação de palavras-chave e sumarização são extraídas de documentos as frases relevantes à elaboração de relatórios toxicológicos. Pela utilização de uma abordagem de sumarização é alcançada uma redução da dimensão dos documentos de 80%, identificando-se corretamente 45 das 53 frases utilizadas numa abordagem convencional, realizada por um especialista do domínio. Já nas abordagens de reconhecimento de entidades, similaridade semântica e identificação de palavras-chave apesar de conseguirem também alcançar ganhos similares obrigam a um maior compromisso no número de frases relevantes identificadas onde, numa abordagem baseada em similaridade semântica, para um ganho de 76% são apenas identificadas 23 das 53 frases utilizadas na abordagem manual, sendo necessário reduzir o ganho a 35% de modo a serem obtidos os mesmos resultados no número de frases corretamente identificadas. Os resultados da avaliação das abordagens são obtidos através de um método de avaliação automático que compara as frases identificadas com as frases de uma abordagem manual.

Palavras-Chave

Processamento de Linguagem Natural, Reconhecimento de Entidades, Extração de Relações, Extração de Informação, Modelos Linguísticos, Transferência de Aprendizagem

Abstract

As technology has evolved in recent years, more and more companies and industries have been moving towards automation of their processes, whether motivated by cost reduction, increased efficiency or speed. The same happens in the linguistic domain, where recent advances bring the comprehension capacity of an artificial intelligence model closer and closer to that of a human being. Thus, taking advantage of the ever-increasing amount of information available online, this work focuses on automatic information extraction from unstructured documents used in chemical toxicology reports. Through entity recognition, semantic similarity, keyword identification and summarization, relevant sentences for toxicological reports are extracted from documents. By using a summarization approach, a reduction in document size of 80% is reached, correctly identifying 45 of the 53 sentences used in a conventional approach, performed by a domain expert. The entity recognition, semantic similarity and keyword identification approaches, despite also achieving similar gains, require a greater compromise in the number of relevant sentences identified where, in an approach based on semantic similarity, for a gain of 76% only 23 of the 53 sentences used in the manual approach are identified, requiring a reduction of the gain to 35% in order to obtain the same results in the number of correctly identified sentences. The results of the evaluation of the approaches are obtained using an automated evaluation method that compares the sentences identified with the sentences from a manual approach.

Keywords

Natural Language Processing, Named Entity Recognition, Relation Extraction, Information Extraction, Language Models, Transfer Learning

Conteúdo

Acrónimos

BERT Bidirectional Encoder Representations from Transformers. 6, 9, 12–14

BioBERT Bidirectional Encoder Representations from Transformers for Biomedical Text Mining. 14

BPE Byte Pair Encoding. i, 6–8

CRF Conditional Random Field. 21

GSC Gold Standard Corpora. 20–22

ML Machine Learning. 18

MLM Masked Language Modeling. 13

NER Named Entity Recognition. 5, 9, 10, 18, 19, 21–23

NSP Next Sentence Prediction. 13

OOV Out Of Vocabulary. 6, 9

PLN Processamento de Linguagem Natural. 2, 5, 14, 22, 27

POS Part-of-Speech. 18, 19

RDF Resource Description Framework. i, 10

RE Relation Extraction. 5, 9, 10

SSC Silver Standard Corpora. 20–22

UMLS Unified Medical Language System. 19

Lista de Figuras

Lista de Tabelas

Capítulo 1

Introdução

Enquadrado no projeto *SafetyDesk - Smart toxicological analysis of chemical substances* do Instituto Pedro Nunes e realizado no âmbito do Estágio/Dissertação do Mestrado em Engenharia e Ciência de Dados da Faculdade de Ciências e Tecnologia da Universidade de Coimbra no ano letivo 2021/2022, o presente documento tem como objetivo a exploração, análise e desenvolvimento de abordagens de Processamento de Linguagem Natural (PLN) que visem a extração de informação em documentos não estruturados.

Na Secção 1.1 é apresentada a motivação para a realização da dissertação, na Secção 1.2 os desafios, na Secção 1.3 os objetivos, na Secção 1.4 as contribuições e, por fim, na Secção 1.5 a estrutura do documento.

1.1 Motivação

A par da evolução tecnológica e da digitalização, os dados têm vindo a ganhar cada vez maior relevância e protagonismo, existindo uma preocupação crescente com o seu armazenamento, tratamento e análise onde toda a informação, quando pública, é hoje acessível a partir de qualquer parte do mundo. Se por um lado existe o benefício de haver mais informação, por outro são necessários os recursos para a sua análise e extração, um processo moroso e muitas vezes impraticável para largas quantidades de dados devido aos gastos inerentes ao tempo despendido.

É o caso da indústria cosmética, a qual estando sob forte escrutínio regulamentar torna necessária a análise das propriedades físico-químicas e toxicológicas de todos os componentes químicos existentes nos produtos antes de estes chegarem ao mercado, tarefa habitualmente realizada por um avaliador de segurança, que tem como função identificar, caracterizar e atestar a conformidade regulamentar de cada substância através da elaboração de um relatório toxicológico. Esta análise envolve em média a consulta de oito bases de dados distintas, algumas das quais com dados não estruturados chegando por vezes, para substâncias menos comuns, a ser necessária a consulta de artigos científicos e relatórios obtidos a partir de motores de pesquisa científica como o PubMed, ScienceDirect e ResearchGate.

Motivado por avanços recentes na área de *PLN*, surge a oportunidade de automatizar o processo referido, com foco na análise de documentos não estruturados, através da pesquisa e recolha automática de dados em múltiplas fontes de informação toxicológicas. Deste modo, para além de se acelerar a construção dos perfis das substâncias será ainda

possível corroborar as conclusões resultantes, através do cruzamento de diversas fontes de informação.

1.2 Desafios

Quando consideramos a automatização da análise de substâncias químicas é incontornável não ter como base a abordagem humana, isto é, o avaliador de segurança, destacando-se durante o processo de análise duas fases: a consulta de bases de dados consideradas fidedignas por experiência do avaliador, habitualmente estruturadas e, na falta de informação suficiente, a consulta de base de dados alternativas, não estruturadas e de conteúdo altamente variável.

No que diz respeito à última fase, o foco deste trabalho, existem vários problemas nomeadamente o elevado número de documentos disponíveis sobre a substância em análise, a dimensão dos documentos e a falta de estrutura. À semelhança de qualquer pesquisa, quando procuramos sobre determinada temática, toxicidade de uma substância química, por exemplo, é inevitável não nos depararmos com resultados indesejados sendo necessária uma filtragem dos documentos devolvidos pela pesquisa, por intermédio de um *ranking*, de modo a excluir ou priorizar cada documento conforme a sua relevância. Já a falta de uma estrutura pré-definida associada à grande dimensão dos documentos levanta dois problemas: o primeiro, originado por limitações algorítmicas, obriga à divisão do documento em segmentos o que, num documento não estruturado, torna difícil a delimitação do texto em segmentos sem que estes percam a coerência e contexto. O segundo, dado pela ausência de uma estrutura conhecida, passa por encontrar em cada documento as partes relevantes à elaboração do relatório toxicológico.

1.3 Objetivos

Atendendo à regulamentação existente na indústria cosmética e à necessidade da elaboração de relatórios toxicológicos, este trabalho tem como objetivo desenvolver uma ferramenta onde, para um conjunto de documentos associados a uma substância química, seja realçada a informação relevante à elaboração do relatório toxicológico da substância em estudo. Desta forma, pretende-se minimizar a quantidade de informação a ser processada por parte do avaliador de segurança.

1.4 Contribuições

Deste trabalho resultaram as seguintes contribuições:

- Elaboração de um módulo utilizado na identificação de frases relevantes de um documento, contando com:
  - Extração de entidades e frases contendo entidades.
  - Extração de frases baseadas em palavras-chave.
  - Extração de frases semanticamente similares.
  - Extração das frases mais relevantes com base em métodos de sumarização.

- Desenvolvimento de uma metodologia de avaliação automática da relevância das frases extraídas.
- Análise dos atributos mais significativos na identificação de frases relevantes.

1.5 Estrutura do Documento

O documento encontra-se dividido nos seguintes seis capítulos:

1. Introdução: Introduz o tema da dissertação passando pela motivação, desafios, objetivos e contribuições.

2. Fundamentos: Aborda as temáticas base necessárias para o entendimento do trabalho.

3. Estado da Arte: Indo de encontro aos objetivos propostos é feito o levantamento de abordagens utilizadas por outros autores.

4. Abordagem: Descrição da arquitetura e funcionamento do programa.

5. Experimentação e Resultados: Apresentação e comparação dos resultados obtidos.

6. Conclusão: Análise crítica dos resultado face aos objetivos propostos.

Capítulo 2

Fundamentos

Ao longo deste capítulo irão ser abordados alguns conceitos base de Processamento de Linguagem Natural (PLN) necessários para o entendimento e realização da dissertação. Sendo o alicerce de qualquer tarefa de PLN, o capítulo inicia-se pela Secção 2.1 referente à tokenização onde se encontram descritos três métodos: Tokenização por palavras 2.1.1, por sub-palavras 2.1.2 e por carateres 2.1.3. Na segunda Secção 2.2 são abordadas duas tarefas de PLN, uma utilizada para o reconhecimentos de entidades - Named Entity Recognition (NER) 2.2.1 e outra para a extração de relações - Relation Extraction (RE) 2.2.2. Já a terceira Secção 2.3 refere-se a modelos utilizados em PLN, com foco em abordagens do estado-de-arte e orientado para as tarefas descritas em 2.2. Por fim a Secção 2.4 faz um apanhado de algumas metodologias de avaliação nomeadamente métodos 2.4.1 tais como treino-teste, treino-validação-teste e validação cruzada e métricas 2.4.2 tais como *accuracy* e *f1-score*.

2.1 Tokenização

Quando lemos uma frase, tomemos esta como exemplo, o cérebro humano executa uma série de operações que permitem a interpretação do texto. Juntamos carateres em palavras, palavras em frases e transformamos frases em informação, um processo que ocorre de forma automática e instantânea. Analogamente, numa abordagem de PLN, existe um conjunto de procedimentos necessários até à interpretação do texto por parte do algoritmo. À divisão do texto em unidades mais pequenas, tokens, chamamos de tokenização e é o primeiro procedimento, comum tanto a abordagens tradicionais como de *Deep Learning*. A tokenização é tanto usada na criação de um vocabulário, baseado num corpus de treino como no pré-processamento de todos os dados introduzidos num modelo, sendo necessário utilizar o mesmo algoritmo em ambos. Entende-se por vocabulário o conjunto de tokens únicos considerados.

A tokenização pode ser dividida em três categorias: Tokenização por palavras, por sub-palavras e por carateres.

2.1.1 Tokenização por Palavras

A tokenização por palavras é o processo mais rudimentar de tokenização e o mais próximo da interpretação humana sendo cada palavra um token. A tokenização por palavras pode consistir na divisão do texto em palavras com base num delimitador, sendo o mais comum

o espaço branco, ou em alternativa, numa abordagem mais elaborada, recorrer a expressões regulares possibilitando lidar com sinais de pontuação, por exemplo.

Exemplo: São 23 horas, devia ir dormir...

Tokenização através de um delimitador

['São, '23', 'horas,', 'devia', 'ir', 'dormir...']

Tokenização através de uma expressão regular

['São, '23', 'horas', 'devia', 'ir', 'dormir']

Este simples processo de tokenização apresenta no entanto desvantagens. À medida que o processo de tokenização acontece, sempre que o algoritmo se depara com uma nova palavra esta é adicionada ao vocabulário o que num corpus de elevada dimensão resulta num vocabulário de elevada dimensão também. Tendo em consideração as implicações de um vocabulário grande, tal como o aumento da complexidade de um modelo, é usualmente imposto ao vocabulário um tamanho máximo a custo da exclusão das palavras menos frequentes - Out Of Vocabulary (OOV). Deste modo, resultante da falta de capacidade de adaptação do vocabulário deixamos de conseguir representar e interpretar as palavras excluídas durante o processo de treino e claro todas as palavras nunca antes encontradas, ou seja, todas as palavras não incluídas no vocabulário. De maneira a que todas as palavras OOV não fiquem sem representação, inclui-se como solução no vocabulário um token especial apelido de UNK, *unknown* vindo do inglês para o qual durante o processo de treino todas as palavras OOV são mapeadas. Apesar de este token especial actuar como substituto de todas as palavras não presentes no vocabulário possibilitando-lhes uma representação, não deixa de se tratar de um e um só token comum a todas as palavras OOV pelo que o significado de cada palavra individualmente não deixa de se perder.

2.1.2 Tokenização por Sub-Palavras

Considerando a dificuldade da tokenização por palavras em lidar com palavras OOV e motivado pela intuição que o significado de uma palavra se pode alcançar através de unidades mais pequenas da própria palavra [33], surge uma abordagem alternativa, a tokenização por sub-palavras. Contrariamente à interpretação inicial que pode surgir baseada no nome, a tokenização por sub-palavras não implica obrigatoriamente a divisão de todas as palavras para a geração de tokens, mantendo na sua forma original as palavras mais frequentes e decompondo apenas as mais raras, menos frequentes, em sub-palavras. Com esta segmentação pretende-se a captação de similaridades sintácticas e semânticas entre palavras e sub-palavras, permitindo interpretar novas palavras e mantendo um tamanho reduzido do vocabulário. Actualmente grande parte dos modelos de estado de arte tal como o Bidirectional Encoder Representations from Transformers (BERT) recorrem a tokenização por sub-palavras.

Existem diferentes métodos de implementação deste processo de tokenização, destacando-se o Byte Pair Encoding (BPE) [33] e WordPiece [39].

Byte Pair Encoding

Tratando-se de um algoritmo de tokenização em sub-palavras, o BPE [33] necessita numa primeira instância da segmentação do texto em tokens 2.1.1, contabilizando de seguida o número de ocorrências de cada token, isto é, a frequência. Tomemos como exemplo o

corpus apresentado em [33], representado na tabela 2.1.

token	ocorrências
low	5
lower	2
newest	6
widest	3

Tabela 2.1: BPE - Corpus inicial.

Cada token é divido em carateres e é acrescentado o caráter </w> ao final de cada token demarcando o seu fim, tabela 2.2.

token	ocorrências
l o w </w>	5
l o w e r </w>	2
n e w e s t </w>	6
w i d e s t </w>	3

Tabela 2.2: BPE - Divisão dos tokens em carateres.

Contendo todos os carateres únicos é criada a primeira instância do vocabulário

Vocabulário = 'd', 'e', 'i', 'l', 'n', 'o', 'r', 's', 't', 'w', '</w>'

De seguida, para cada token, os carateres são agrupados dois a dois, contabilizando-se a frequência de cada par de carateres. No caso de l o w </w> tem-se lo ow e w</w> como pares de carateres. A contabilização da frequência é o resultado da soma do número de ocorrências de cada par de carateres ao longo de todos os tokens. Conhecido o par mais frequente os dois carateres são concatenados num só e o par, agora transformado em um caráter é adicionado ao vocabulário. Em caso de empate qualquer par pode ser escolhido, 2.3 e 2.4.

token	ocorrências
l o w </w>	5
l o w e **r** **</w>**	2
n e w **e s** t </w>	6
w i d **e s** t </w>	3

Tabela 2.3: BPE - Alguns pares de carateres na primeira iteração.

par	ocorrências
lo	7 (5+2)
r</w>	2
es	9 (6+3)

Tabela 2.4: BPE - Contabilização de alguns pares de carateres da tabela 2.3.

É adicionado ao vocabulário o par de caráter mais frequente.

Vocabulário = 'd', 'e', 'i', 'l', 'n', 'o', 'r', 's', 't', 'w', '</w>, es'

token	ocorrências
l o w </w>	5
l o w e r </w>	2
n e w es t </w>	6
w i d es t </w>	3

Tabela 2.5: BPE - Transformação do par de carateres mais frequente num só caráter.

Repetem-se sucessivamente os passos enumerados até que se alcance uma de duas condições de paragem. A impossibilidade de continuar a agrupar pares de carateres dois a dois ou o tamanho máximo do vocabulário, definido à priori, ser atingido. Como a primeira instância do vocabulário é criada a partir de todos os carateres únicos, no limite apenas haverá um token mapeado para UNK caso esse token, obrigatoriamente um caráter, não tenha aparecido durante o treino. Atendendo também ao processo de construção do vocabulário as palavras mais comuns acabam por ter representação integra através de só um token, já que a concatenação dos pares de carateres é feita com base na frequência o que leva eventualmente à formação de palavras completas.

WordPiece

Inicialmente desenvolvido para resolver um problema de segmentação Japonês/Coreano no âmbito de um sistema de reconhecimento de voz [32], o WordPiece [39] é também um algoritmo de tokenização em sub-palavras. Praticamente idêntico ao BPE o WordPiece difere no método de seleção do par de carateres adicionado ao vocabulário. Enquanto que no BPE o par é determinado em função da probabilidade, no WordPiece a escolha baseada no maior resultado da divisão da frequência do par pelo produto das frequências de cada uma das suas partes. Tomemos como exemplo o par e s:

$$Resultado = \frac{\#\text{ (e s)}}{\#\text{(e)} * \#\text{(s)}}$$

Unigramas, Bigramas e N-Gramas

Unigramas, bigramas e N-gramas são sequências de um, dois ou N carateres ou de uma, duas ou N palavras utilizados para prever o próximo token de uma sequência com base em conhecimento adquirido, representado através de probabilidades.

Assume-se o seguinte corpus e os seus bigramas:

"Vim escrever a tese no DEI--> ["Vim escrever", "escrever a", "a tese", "tese no", "no DEI"]

"Amanhã vou escrever a tese no DEI--> ["Amanhã vou", "vou escrever", "escrever a", "a tese", "tese no", "no DEI"]

"Ontem gostava de ter estado no DEEC--> ["Ontem gostava", "gostava de", "de ter", "ter estado", "estado no", "no DEEC"]

Uma aplicação direta do bigrama é, por exemplo, o cálculo da probabilidade de um token se suceder a outro:

$$P(t_n|t_{n-1}) = \frac{\#(t_{n-1},t_n)}{\#(t_{n-1})}$$

2.1.3 Tokenização por carateres

Semelhante à tokenização por sub-palavras, podendo ser considerado um caso mais simples, temos a tokenização por carateres. Partindo da premissa que a língua inglesa ou qualquer outra língua tem um número de carateres muito inferior ao número de palavras, a tokenização por carateres como o nome indica segmenta o texto em carateres, destacando-se nesta abordagem duas vantagens e duas desvantagem. A primeira vantagem vai de encontro à motivação do processo de tokenização, o tamanho do vocabulário, que sendo apenas constituído por carateres não só é consideravelmente mais pequeno para um mesmo corpus de treino quando comparado com um processo de tokenização por palavras ou sub-palavras, que à partida por maior que venha a ser o tamanho do corpus o vocabulário não aumenta ou caso aumente a diferença é desprezível para a performance do modelo. Em segundo lugar e tal como na tokenização por sub-palavras existe a vantagem de lidar com palavras OOV. Relativamente às desvantagens, como os tokens são exclusivamente carateres, para representar uma frase são necessários vários tokens o que em alguns modelos como é o caso do BERT poderá constituir uma limitação já que este processo leva à geração de sequências extensas. No que toca a informação a representação de uma palavra por intermédio de um conjunto de carateres dá ao modelo uma quantidade útil de informação inferior que a própria palavra em si ou pela subdivisão como visto na secção anterior.

2.2 Tarefas

Enquadrado no objetivo deste trabalho e considerando as várias tarefas de extração de informação disponíveis, assumiram-se as tarefas de reconhecimento de entidades - NER e de extração de relações - RE. Tanto uma como a outra desempenham papéis fulcrais em qualquer tarefa de extração de informação, onde a extração de entidades permite, para cada frase, perceber quais as entidades envolventes e a extração de relações entender de que forma estas entidades se relacionam.

2.2.1 Named Entity Recognition

NER é uma tarefa de extração de informação que tem como objectivo identificar e categorizar entidades presentes em texto não estruturado em categorias tais como pessoas, organizações e localizações ou em certas circunstâncias entidades mais específicas como reagentes químicos e moléculas, permitindo uma análise rápida e clara das entidades envolvidas em cada segmento do texto. Deste modo, quer seja para utilização do um humano, servindo de elemento facilitador de procura pela anotação das entidades, evitando que este necessite de fazer uma análise extensa do documento para encontrar partes relevantes ou em conjugação com outras tarefas como será falado na próxima subsecção, a tarefa de NER revela-se extremamente útil.

Como nem sempre uma entidade corresponde a uma só palavra, como é o caso de um nome ou uma localização, torna-se difícil por vezes definir onde começa e acaba uma entidade. Uma abordagem para identificação de entidades é a BIO tagging [30] que considerando o problema apresentado em vez de tentar categorizar uma ou várias palavras de uma só vez, categoriza cada frase palavra a palavra onde a primeira palavra de uma entidade é anotada com B, as seguintes com I e com O todas as palavras que não tenham entidade associada.

2.2.2 Relation Extraction

RE é uma tarefa de extração de informação que tem como objectivo transformar informação textual não estruturada em informação estruturada, através da extração de relações semânticas entre entidades tais como pessoas, organizações, localizações e entidades geopolíticas. Considerando que as relações/associações são feitas relativamente a entidades e não às palavras em si, a tarefa de RE subentende NER na sua base.

Apesar da existência de *datasets* públicos tais como o DBpedia e o Wikidata que disponibilizam relações entre entidades como por exemplo pessoas e sua nacionalidade ou pessoas e a sua localização, representados através de Resource Description Framework (RDF) triples tabela 2.6, ou seja, tuplos no formato de entidade-relação-entidade, estas situações são usualmente generalistas não se aplicando a todos os cenários, nomeadamente domínios específicos. Como tal, são apresentadas cinco tipologias de algoritmos de RE: Padrões, supervised machine learning, semi-supervived via bootstrapping or distant supersivion e unsupervised.

Entity	Relation	Entity
Injury	disrupts	Physiological Function
Pharmacologic Substance	treats	Pathologic Function

Tabela 2.6: RDF triples (adaptado de Speech and Language Processing (3rd ed. draft).

Padrões

Começando pela mais simples e antiga abordagem de extração de relações, temos a extração de relações através de padrões, apresentada por [14] em 1992. Utilizada para inferir relações de hiponímia inicialmente, a extração de relação por padrões foca a atenção num conjunto empírico de padrões léxico-sintáticos a partir dos quais se extrai relações. Desta forma, apesar da exaustividade necessária para formular um conjunto amplo e alargado de padrões, como são criados manualmente é possível ajustá-los a domínios específicos possibilitando uma alta precisão a troco de um baixo *recall*.

Supervised Machine Learning

Tal como em qualquer abordagem de aprendizagem supervisionada, a extração de relações por aprendizagem supervisionada requer também dados anotados sendo necessário definir à priori as relações e entidades através da anotação dos dados de treino, caso não estejam já anotados. Uma das vantagens desta abordagem é a facilidade de encontrar pares de entidades e descobrir diversos tipos de relações entre as mesmas. Caso pretendamos acelerar o processo do modelo, temos ainda a flexibilidade de adicionar um passo intermédio, onde o modelo, em vez de partir de imediato para a procura de pares de entidades ou relações primeiro implementa uma classificação binária onde verifica se existe alguma relação entre pares de entidades. Apesar dos nítidos pontos positivos, a aprendizagem supervisionada sofre de um calcanhar de aquiles, não sendo sempre possível recorrer a esta abordagem considerando que pode não haver dados anotados em quantidade suficiente nem ser possível a sua anotação devido ao tempo, custos e incerteza associada. No entanto, caso efetivamente tenhamos dados anotados e apesar da falta de capacidade de generalização, a aprendizagem supervisionada pode revelar-se uma boa escolha.

Bootstrapping

Como visto na secção anterior, nem sempre existem dados anotados em quantidade suficiente ou é viável a sua anotação para o treino de um modelo através de uma abordagem supervisionada. Assumindo que temos apenas ao nosso dispor uma pequena quantidade inicial de dados anotados, sejam entidades ou padrões, apelidemos de *"seeds"*, podemos a partir desses dados encontrar mais entidades e padrões de forma sucessiva, aumentado o número de dados anotados. No caso de as *seeds* serem entidades, procuramos em vários documentos frases em que estas ocorram, generalizando o seu contexto e extraindo padrões de maneira a encontrar novas entidades, figura 2.1. A esta abordagem semi-supervisionada chama-se de *bootstrapping*.

function BOOTSTRAP(*Relation R*) **returns** *new relation tuples*

tuples ← Gather a set of seed tuples that have relation *R*
iterate
sentences ← find sentences that contain entities in *tuples*
patterns ← generalize the context between and around entities in *sentences*
newpairs ← use *patterns* to grep for more tuples
newpairs ← *newpairs* with high confidence
tuples ← *tuples* + *newpairs*
return tuples

Figura 2.1: Algoritmo de *bootstraping* para aprender relações (retirado de Speech and Language Processing (3rd ed. draft).

Distant Supervision

Distant supervision [23] é também uma abordagem semi-supervisionada que combina uma abordagem supervisionada com *bootstrapping* onde, adquirido a partir de um *dataset*, é utilizado um grande número de *seeds*. Recorrendo a um largo conjunto de textos e um anotador de entidades são anotadas todas as entidades presentes nos textos e selecionadas as frases cujas entidades anotadas correspondem às entidades presentes nas *seeds*. Retirado de Speech and Language Processing (3rd ed. draft) segue-se um exemplo.

Considerando (<Edwin Hubble, Marshfield> e <Albert Einstein, Ulm>) como duas *seeds* da relação de local-de-nascimento encontram-se as seguintes frases:

- ...**Hubble** was born in **Marshfield**...
- ...**Einstein**, born (1879), **Ulm**...
- ...**Hubble's** birthplace in **Marshfield**...

A partir das frases obtidas é criado um *dataset* de treino onde cada entrada é dada na forma de <relação, entidade1, entidade2>:

- <born-in, Edwin Hubble, Marshfield>
- <born-in, Albert Einstein, Ulm>

- <born-year, Albert Einstein, 1879>

Por último, aplica-se uma abordagem de aprendizagem supervisionada no *dataset* gerado.

function DISTANT SUPERVISION(*Database D, Text T*) **returns** *relation classifier C*

foreach relation *R*
  foreach tuple (*e1,e2*) of entities with relation *R* in *D*
    sentences ← Sentences in *T* that contain *e1* and *e2*
    f ← Frequent features in *sentences*
    observations ← observations + new training tuple (*e1, e2, f, R*)
C ← Train supervised classifier on *observations*
return *C*

Figura 2.2: Algoritmo de *distant supervision* para extração de entidades (retirado de Speech and Language Processing (3rd ed. draft).

Unsupervised

Por fim, como última abordagem de extração de relações temos uma abordagem não supervisionada. À semelhança de qualquer outra abordagem não supervisionada esta não requer dados anotados, tornado-a ideal em cenários onde a anotação não é possível. Apesar de conseguir gerar associações entre diversas relações, muitos dos algoritmos existentes actualmente focam-se em relações expressadas através de verbos pelo que é provável que relações expressadas nominalmente passem despercebidas a abordagens não supervisionadas.

2.3 Modelos

Aproveitando o conhecimento linguístico adquirido de modelos pré-treinados (*transfer learning*, nesta secção são abordados 3 modelos linguísticos: 2.3.1. BERT, 2.3.2. BioBERT e 2.3.3. SciBERT, sendo os últimos dois variações do primeiro. Todos estes modelos fazem uso de *transfer learning*, beneficiando do pré-treino em corpus grandes e diversificados. Assim, para uma mesma tarefa um modelo linguístico pré-treinado consegue mais facilmente entender os dados de treino, o que se transmite em melhores resultados e numa necessidade menor de dados de treino face a um modelo treinado de raiz.

2.3.1 BERT

Introduzido recentemente por investigadores da Google, o BERT [6], é um modelo pré-treinado com recurso à Wikipédia Inglesa e a um *BooksCorpus* [40] contando com 2500 milhões e 800 milhões de palavras respetivamente. Contrariamente a modelos unidirecionais que interpretam o texto sequencialmente da esquerda para direita ou da direita para a esquerda, ou de modelos que fazem uma concatenação da interpretação sequencial em ambos os sentidos, isto é, da esquerda para a direita e da direita para esquerda, o BERT é naturalmente bidirecional possibilitando a melhor representação contextual de cada palavra. Destacam-se duas etapas principais no BERT: Pré-treino e *Fine-tuning*

Pré-treino

Uma vez que um modelo bidirecional não pode ser treinado como um modelo de linguagem normal, o BERT é pré-treinado utilizando duas tarefas não supervisionadas: *Masked Language Modeling (MLM)* e *Next Sentence Prediction (NSP)*.

MLM é uma tarefa de modelação de linguagem, usada para representação textual, onde, para cada sequência de palavras, algumas das palavras são mascaradas com [MASK], e o modelo tenta fazer a previsão da [MASK] com base nas palavras circundantes. É através desta tarefa que o BERT consegue ser um modelo bidirecional.

NSP é responsável por fazer o BERT compreender a relação entre fases. Esta relação é alcançada através de um mecanismo de previsão onde o modelo, recebendo pares de frases de um documento, aprende a prever se a segunda frase é a continuação da primeira.

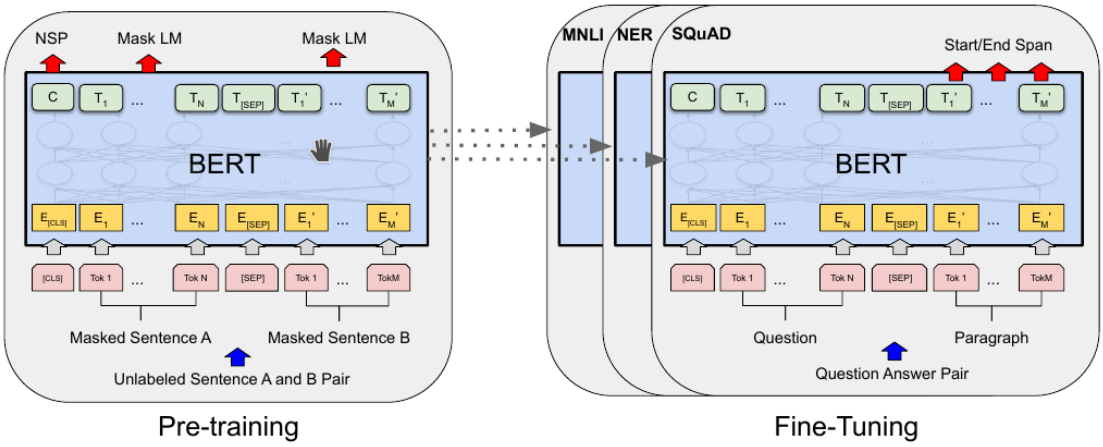


Figura 2.3: Arquitetura *pre-training* e *fine-tuning* na tarefa de perguntas e respostas. Retirado de [6].

Fine-tuning

O processo de *fine-tuning* é o processo de pequenos ajustes no modelo, adaptando-o a uma tarefa e domínio específico designados por *downstream task*. Ao fazer *fine-tuning* estamos a aproveitar o amplo conhecimento linguístico adquirido pelo modelo no pré-treino, o que possibilita resultados de estado de arte numa vasta maioria de tarefas linguísticas com apenas uma reduzida quantidade de dados e um baixo tempo de treino.

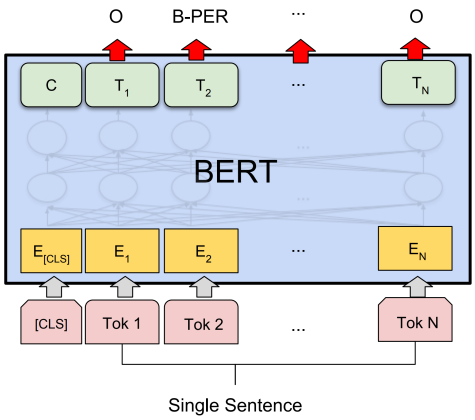


Figura 2.4: Arquitetura *fine-tuning* na tarefa de reconhecimentos de entidades. Retirado de link e consultado em 24/08/2022.

2.3.2 BioBERT

Apesar dos avanços no estado de arte levados a cabo pelo BERT [6] em tarefas de PLN, inclusivamente na área biomédica, com o aumento do volume da literatura a um ritmo médio diário de 3000 novos artigos, surge a necessidade de combater a principal limitação do modelo no domínio biomédico, comum também a modelos do estado de arte anteriores - o corpus de treino.

Motivado pelo problema enunciado surge o Bidirectional Encoder Representations from Transformers for Biomedical Text Mining (BioBERT). Inicializado com os mesmos pesos do BERT, o BioBERT [20] complementa o pré-treino do seu antecessor com dados do domínio biomédico nomeadamente 4,5 biliões americanos de palavras provenientes de *abstracts* do PubMed e 13,5 biliões americanos de palavras de artigos completos do PMC, apresentando-se na sua versão final como um modelo pré-treinado num corpus de linguística geral e específica do domínio biomédico.

2.3.3 SciBERT

Partilhando a mesma motivação do BioBERT, o SciBERT [4] surge igualmente no sentido de colmatar a falta de modelos pré-treinados em corpus biomédicos.

Utilizando a mesma arquitetura do BERT, o SciBERT difere do modelo original em dois aspetos: vocabulário e corpus de pré-treino. Enquanto que o BERT utiliza *WordPiece* [39] para a tokenização do input e consequente criação do vocabulário o SciBERT usa *SentencePiece* [18] baseada em corpus científicos para a construção do vocabulário e treinado com 1,14 milhões de artigos científicos oriundos do corpus da Semantic Scholar [2].

2.4 Metodologias de Avaliação

Quando treinamos um modelo fazemo-lo porque acreditamos ser mais eficiente, mais económico ou até mesmo mais fiável que um ser humano numa determinada tarefa, pretendendo que obtenha os melhores resultados e que faça jus às expectativas. Nesta secção irão ser abordados métodos e métricas de avaliação do desempenho de um modelo.

2.4.1 Métodos de Avaliação

No processo de avaliação de um modelo, dependendo da quantidade de dados disponível e da sua tipologia são vários os procedimentos de avaliação pelos quais podemos optar, estando a escolha do procedimento subjacente à conciliação entre as características de cada modelo e dos dados. Serão abordados procedimentos como Treino-Teste (TT), Treino-Validação-Teste (TVT) e Validação Cruzada vindo do inglês *Cross Validation* (CV).

Treino-Teste

No procedimento de TT, o mais rudimentar dos três apresentados, dividimos os dados em dois blocos, um de treino de um de teste, atribuindo uma percentagem a cada bloco. Em regra geral é dado ao bloco de treino 70% dos dados e ao bloco de teste 30%, totalizando a percentagem da soma blocos sempre 100%. O bloco de treino é utilizado exclusivamente para treino e o bloco de teste exclusivamente para teste não podendo estes dados fazer parte de qualquer processo de treino. Pretende-se com isto simular a utilização do modelo no mundo real com dados nunca antes vistos. Caso utilizemos os dados de teste no treino, não só não vamos estar a testar com dados nunca antes vistos, desvirtuando a tarefa de teste de um cenário mais aproximado ao mundo, real como corremos o risco de o modelo se adaptar aos dados em que foi treinado desvirtuando o teste.

Figura 2.5: Treino (70%) e Teste(30%).

Treino-Validação-Teste

O TVT é de certa forma muito semelhante ao TT onde os blocos de treino e teste servem exactamente o mesmo propósito. A diferença encontra-se porem na utilização de um bloco complementar, o bloco de validação. Através do bloco de validação o desempenho do modelo é testado ajustando-se de forma iterativa os parâmetros do modelo, impedindo o sobretreino e melhorando o seu desempenho. Após os parâmetros estarem ajustados o modelo é testado no bloco de teste. Utiliza-se normalmente 60% para os dados de treino, 20% para os dados de validação e 20% para os dados de teste.

Figura 2.6: Treino (60%), Validação (20%) e Teste(20%).

Teste Cruzado

Já o teste cruzado difere bastante de o TT e TVT. Em vez dividir os dados em treino-teste ou treino-validação-teste, divide-os em K blocos de dimensão idêntica, usemos K=5, onde 4 dos 5 blocos atuam como treino e um como teste. O bloco de teste vai alternando até que todos os blocos tenham sido usados como teste.

Teste	Treino	Treino	Treino	Treino
Treino	Teste	Treino	Treino	Treino
Treino	Treino	Teste	Treino	Treino
Treino	Treino	Treino	Teste	Treino
Treino	Treino	Treino	Treino	Teste

Figura 2.7: Teste cruzado com K=5.

2.4.2 Métricas de Avaliação

Tal como abordado anteriormente com os métodos de avaliação, as métricas de avaliação também em si diferem, sendo algumas mais adequadas que outras mediante certas situações. Serão apresentadas 4 métricas consideradas *strandart*: *Accuracy*, *precision*, *recall* e *f1-score*.

Accuracy

A *accuracy* é uma métrica de avaliação do desempenho de um modelo, utilizada em tarefas de classificação.

$$Accuracy = \frac{\#PrevisoesCorretas}{\#TotalPrevisoes}$$

Apesar de a *accuracy* ser uma métrica aceitável em diversas situações, como o caso de uma classificação binária num *dataset* balanceado, é necessário ter atenção a *datasets* não balanceados. Colocando o exemplo de um *dataset* com 99% dos dados da classe A e somente 1% da classe B, ao usarmos a métrica da *accuracy*, o modelo ao classificar todos os dados como sendo da classe A obteria uma *accuracy* de 99%. Apesar de 99% aparentar ser um excelente resultando não o é pois o modelo estaria a falhar a classificação de todos os dados da classe B.

Matriz de Confusão

Uma matriz de confusão, é uma matriz de tamanho N x N usada para analisar o desempenho de modelos em tarefas de classificação onde N representa o número de classes existentes. Esta matriz tem como base a comparação entre o output do modelo, ou seja, a classe prevista e classe real, introduzindo-nos a conceitos como Verdadeiros Positivos (VP), Verdadeiros Negativos (VN), Falsos Positivos (FP), Falsos Negativos (FN) e também *Precision*, *Recall* e *F1-Score*. A seguinte figura 2.8 demonstra uma matriz de confusão para uma tarefa de classificação binária.

		Classe Prevista	
		Positiva	Negativa
Classe Real	Positiva	VP	FN
	Negativa	FP	VN

Figura 2.8: Matriz de confusão com N=2.

Onde:

- VP - O modelo classificou como positivo e é positivo.
- VN - O modelo classificou como negativo e é negativo.
- FP - O modelo classificou como positivo e é negativo.
- FN - O modelo classificou como negativo e é positivo.

Accuracy

A *accuracy* também pode ser calculadas recorrendo a uma matriz de confusão:

$$Accuracy = \frac{TP+TN}{VP+VN+FP+FN}$$

Precision

A *precision* é dada pela relação entre as observações positivas corretas (TP) e o total de observações classificadas como positivas (TP + FP).

$$Precison = \frac{TP}{TP+FP}$$

Recall

O *recall* é dado pela relação entre as observações positivas corretas (TP) e o total de observações que são realmente positivas (TP + FN)

$$Recall = \frac{TP}{TP+FN}$$

F1-Score

Como usualmente quanto maior a *precision* menor o *recall* e quanto maior o *recall* menor a *precision*, surge o *F1-score*, calculado pela média harmónica entre a *precision* e o *recall*.

$$F1 = 2 * \frac{Precision*Recall}{Precision+Recall}$$

Capítulo 3

Estado da Arte

Ao longo deste capítulo serão apresentados trabalhos relacionados com extração de informação, mais concretamente na tarefa de Named Entity Recognition (NER) 2.2.1 e em documentos não estruturados do domínio científico. Apesar de o domínio científico poder ser considerado muito lato, nomeadamente no que toca às terminologias utilizadas dentro de cada área, os desafios e obstáculos existentes para as tarefas propostas são gerais, pelo que nesta secção são tidas em consideração abordagens de todo o domínio cientifico, beneficiando-se assim da análise de um conjunto mais alargado de estudos.

O capítulo encontra-se dividido em quatro secções, cada uma referente de abordagem diferente. A Secção 3.1 é referente a modelos baseados em regras, a Secção 3.2 a modelos baseados em dicionários, a Secção 3.3 a modelos de Machine Learning (ML) e, por último, a Secção 3.4 tem como foco modelos linguísticos.

3.1 Modelos Beaseados em Regras

Os modelos baseados em regras, como o nome indica, regem-se por um conjunto de regras, recorrendo regularmente ao uso de expressões regulares para a criação das regras. Tratando-se de um processo manual envolve uma forte componente humana e de conhecimento específico, não sendo possível a generalização para outros ramos. Devido à regularidade com que novas terminologias e expressões aparecerem, para além do esforço humana inicial os modelos baseados em regras requerem um acompanhamento constante para que para novos documentos as relações continuem a conseguir ser captadas.

Em [27] é desenvolvido um conjunto de regras baseado na linguística computacional e que incorporam informação semântica relativa ao domínio da alimentação. O corpus utilizado é o *FoodBase* e consiste em 1000 receitas anotadas, extraídas do site Allrecipes, ao longo de 5 categorias, 200 receitas para cada: *Appetizers and snacks*, *Breakfast and Lunch*, *Dessert*, *Dinner*, e *Drinks*. A criação do modelo baseia-se em 4 etapas:

1. Pré-processamento do texto não estruturado como pela remoção de carateres "não standard"e espaços brancos em duplicado.
2. Análise morfológica através da combinação dos resultados de dois Part-of-Speech (POS) *taggers* a fim de o obter "anotações"mais robustas.
3. Criação das regras com base em caraterísticas semânticas do domínio. É nesta fase que são identificadas as possíveis frases candidatas a entidades.

4. Classificação das frases candidatas como entidades de comida ou não entidades

Apesar do modelo apresentar bons resultados, alcançando um *f1-score* de 96,05%, tem a desvantagem de não se poder generalizar para outras áreas já que as regras utilizadas foram criadas de acordo com propriedade léxicas e semânticas do domínio da alimentação.

3.2 Modelos Baseados em Dicionários

Utilizando um conjunto pré-definido de tokens-entidades manualmente anotados, uma abordagem baseada em dicionários recorre uma tabela de procura para associar um token ou conjunto de tokens a uma entidade.

Em [31] é apresentado um sistema modular - cTAKES, baseado em dicionários para a tarefa de extração de entidades e construído com recurso a tecnologias *open-source* tais como o OpenNLP toolkit. O sistema é constituído por 7 componentes que funcionam sequencialmente, existindo uma dependência da componente seguinte pela anterior, com exceção da primeira:

1. *Sentence boundary detector*: Divide o texto em frases, decidindo o que é considerado uma frase, isto é, onde começa e acaba cada através do uso de pontos finais, pontos de interrogação e pontos de exclamação, entre outros.

2. *Tokenizer*: Para cada frase o tokenizer separa a frase por espaços e pontuação.

3. *Normalizer*: Proporciona uma representação textual alternativa às palavras de acordo com as suas propriedade léxicas tais como contrações (can't –> can not), variantes de escrita (color –> colour), maiúsculas (Evidence –> evidence) e inflexão (shows –> show e showed –> show), permitindo mapear múltiplas variações da mesma palavra com representações textuais diferentes numa só.

4. *POS tagger*: Associa cada token do input a uma categoria léxica tais como substantivos, verbos, adjetivos ou advérbios.

5. *Shallow parser*: De acordo com as categorias lexicais atribuídas, agrupa os tokens em categorias hierárquicas superiores.

6. *NER annotator*: Utiliza um algoritmo de procura baseado em dicionários numa janela de procura do tamanha de cada substantivo ou conjunto de substantivos presentes na frase de input associando-os a entidades. O dicionário utiliza um subset do Unified Medical Language System (UMLS) [25], enriquecido com sinónimos de uma lista de terminologias da UMLS e Mayo.

7. *Negation annotator*: Determina se uma entidade está a ser negada através de uma abordagem baseada em regras.

Para a componente de NER, o cTAKES apresenta um *f1-score* de 0,715 para correspondências exatas e um *f1-score* de 0,824 para correspondências sobrepostas. Considera-se uma correspondências exata quando o intervalo detetado pelo *sentence boundary* corresponde exatamente às anotações e uma correspondência sobreposta quando apenas corresponde parcialmente. Apesar dos resultados satisfatórios é de notar que manter o dicionário atualizado é um procedimento dispendioso. Já para o *negation annotator* obteve-se um *f1-score* de 0,94 [31].

3.3 Machine Learning

Dispensando a necessidade da intervenção de um especialista na criação de regras e dicionários, surgem os modelos de *machine learning*. Estes modelos, em vez de se regerem por um conjunto de regras ou dicionários aprendem de forma automática, o que possibilita também o uso da mesma arquitetura em domínios diferentes.

Considerando os obstáculos existentes na anotação de dados de domínios específicos, tais como a necessidade de mão de obra altamente especializada e a extensividade do processo de anotação, [11] estuda o impacto do uso de *transfer learning*, através de uma LSTM-CRF, numa tarefa de reconhecimento de entidades biomédicas. A avaliação é feita com recurso a 23 *datasets* de alta fidedignidade - Gold Standard Corpora (GSC) (Apêndice A: *Gold Standard Corporas* tabela 1), manualmente anotados por especialistas para uma de 4 entidades distintas: *chemicals*, *diseases*, *species* e *genes/proteins*. O modelo é treinado individualmente para cada GSC, com e sem *transfer learning*, utilizando treino-validação-teste com 60%, 10% e 30% respetivamente. Para o pré-treino da LSTM-CRF utlizado na abordagem de *transfer learning* é criado um dataset de forma automática - Silver Standard Corpora (SSC) para cada entidade, selecionando-se aleatoriamente 50000 de 174999 *abstracts* do CALBC-SSC-III-Small corpus [15]. Os resultados mostram-se promissores para a utilização de *transfer learning*, alcançando-se melhorias em 22 dos 23 GSC face apenas ao treino no *target* dataset. A Tabela 3.1 estabelece uma comparação entre a abordagem sem *transfer learning* - *baseline* (B) e de *transfer learning* (TL) para cada entidade.

	Precision (%)		Recall (%)		F1-score (%)	
	B	TL	B	TL	B	TL
Chemicals	**87,10**	87,05	89,19	**89,47**	88,08	**88,21**
Diseases	80,41	**81,41**	81,13	**82,46**	80,73	**82,09**
Species	84,18	**84,52**	84,44	**90,12**	84,20	**87,01**
Genes/proteins	82,09	**83,38**	80,85	**83,08**	81,20	**83,09**

Tabela 3.1: *Precision*, *recall* e *f1-score* em B e TL (retirado de [11]).

Realçam-se os benefícios de *transfer learning* em *target* datasets mais pequenos onde gradualmente, quanto maior o dataset, menores são as diferenças entre a abordagem *baseline* e de *transfer learning*, tornando este método particularmente promissor em situações onde a quantidade de dados anotados disponíveis é mínima.

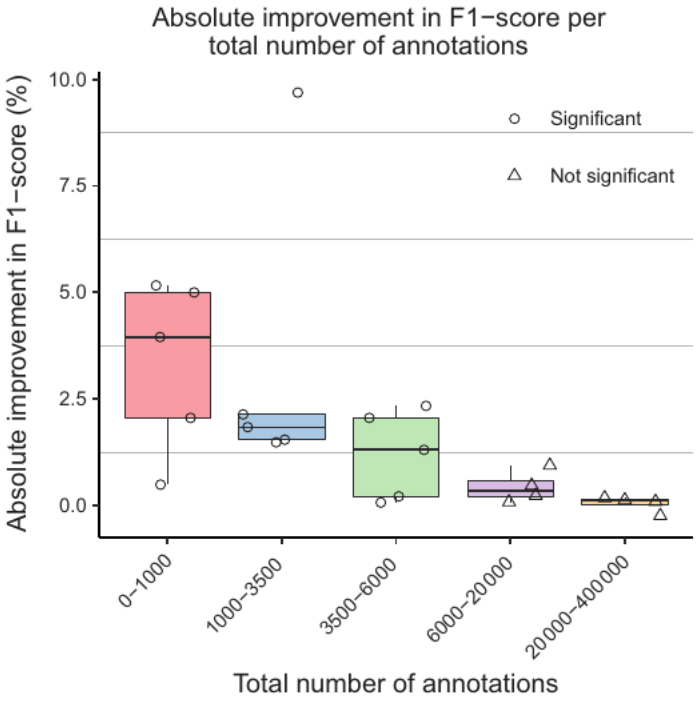


Figura 3.1: Boxplot da evolução do *f1-score* em função do número de dados anotados (retirado de [11]).

Dando seguimento à exploração em [11], analisando também a influência do uso de um modelo pré-treinado numa tarefa de NER, [37] compara os resultados de uma LSTM-CRF pré-treinada num GSC, com uma LSTM-CRF pré-treinada num SSC e com uma LSTM-CRF sem pré-treino, treinada apenas no *target* dataset. Os GSC utilizados no pré-treino são obtidos através da concatenação, por entidade, de 34 GSC. Consideram-se as seguintes 5 entidades: *chemicals*, *diseases*, *species*, *genes/proteins* e *cell lines*. Para a criação dos SSC é utilizado um Conditional Random Field (CRF) em cada um dos os 5 GSC, aplicando-se o modelo resultante em todos os *abstracts* do PubMed até 2015, excluindo todas as entidades anotadas cuja probabilidade dada pelo CRF se encontre abaixo de um determinado *treshold*.

	GSTP			SSPT		
	Sentences	Annotations		Sentences	Annotations	
Entity type		Total	Unique		Total	Unique
Chemicals	281 883	287 972	70 370	1 286 274	1 539 718	78 822
Diseases	205 042	29 929	4799	799 849	850 376	48 162
Species	26 357	4772	1141	318 819	326 474	17 459
Gene/protein	88 863	87 015	19 985	323 518	378 707	50 149
Cell line	19 853	7000	2105	4577	4636	2121

Tabela 3.2: Estatísticas dos GSC e SSC utilizados no pré-treino dos modelos (retirado de [37]).

		Precision (%)			Recall (%)			F1-score (%)		
		No	GSPT	SSPT	No	GSPT	SSPT	No	GSPT	SSPT
Cell Lines	JNLPBA	69,46	65,69	**73,02**	**65,35**	64,10	62,04	**67,34**	64,88	67,09
	CellFinder	89,77	82,81	**91,57**	60,77	**81,54**	58,46	72,48	**82,17**	71,36
	CLL	75,29	76,06	**82,19**	**83,12**	70,13	77,92	79,01	72,97	**80,00**
	Gellus	87,34	82,89	**88,78**	55,87	62,75	**73,68**	68,15	71,43	**80,53**
Chemicals	CHEMDNER patent	82,30	83,21	**84,15**	**89,10**	87,90	88,64	85,56	85,49	**86,34**
	CHEBI	75,43	**83,31**	79,81	77,47	**77,60**	77,48	76,44	**80,36**	78,63
	BioSemantics	82,10	80,80	**84,51**	84,41	**84,61**	82,13	83,24	82,66	**83,30**
	CHEMDNER	89,25	**90,51**	88,72	87,65	87,30	**87,83**	88,44	**88,87**	88,28
	CDR	91,89	**93,51**	91,75	89,59	92,32	**92,45**	90,72	**92,92**	92,10
	SCAI Chemicals	72,57	75,00	**78,20**	65,60	**80,80**	76,53	68,91	**77,79**	77,36
Diseases	BioSemantics	**75,90**	73,31	72,75	78,94	82,37	**82,60**	77,39	**77,58**	77,36
	CDR	**83,67**	83,51	81,08	83,26	**83,90**	83,26	83,46	**83,71**	82,15
	Variome	87,08	**87,23**	86,38	78,65	77.83	**79,60**	82,65	82,26	**82,85**
	miRNA	79,23	81,38	**81,93**	77,33	**83,33**	83,03	78,27	82,34	**82,48**
	NCBI Disease	85,28	85,93	**86,81**	83,23	84,81	**85,06**	84,24	85,37	**85,92**
	SCAI Disease	75,28	**81,19**	77,91	75,49	**79,01**	75,49	75,39	**80,09**	76,68
Genes	CHEMDNER patent	64,55	65,69	**66,40**	77,10	**79,14**	78,58	70,27	71,79	**71,98**
	BioCreative II GM	77,96	**79,95**	79,55	**77,95**	75,92	77,60	77,96	77,88	**78,56**
	JNLPBA	**80,57**	79,12	80,18	83,00	81,94	**84,45**	81,77	80,51	**82,26**
	CellFinder	**83,56**	83,33	77,09	66,22	74,47	**76,06**	73,89	**78,65**	76,57
	OSIRIS	84,00	84,57	**84,67**	79,38	**90,38**	87,29	81,63	**87,38**	85,96
	DECA	65,61	**66,05**	64,57	72,52	71,85	**73,99**	68,90	68,82	**68,96**
	Variome	**93,45**	93,04	92,40	95,01	95,19	**95,47**	**94,23**	94,11	93,91
	FSU-PRGE	87,39	**87,97**	87,36	88,73	**88,84**	88,52	88,05	**88,40**	87,94
	IEPA	**87,85**	80,77	87,54	84,33	84,00	**91,33**	86,05	82,35	**89,40**
	BioInfer	85,81	**85,97**	85,05	82,21	83,30	**86,77**	83,97	84,61	**85,90**
	miRNA	**71,24**	64,43	67,82	57,04	**75,95**	73,88	63,36	69,72	**70,72**
	Loctext	**82,17**	79,52	81,97	74,81	74,63	**83,96**	78,32	77,00	**82,95**
Species	CellFinder	78,63	**84,48**	80,51	86,79	**92,45**	89,62	82,51	**88,29**	84,82
	miRNA	**93,30**	91,89	90,78	85,90	**89,87**	82,38	89,45	**90,87**	86,37
	s800	72,20	**75,15**	73,02	70,11	70,11	**70,30**	71,14	**72,54**	71,63
	Loctext	**94,19**	92,63	93,55	88,04	**95,65**	94,57	91,01	**94,12**	94,05
	Linneaus	92,09	94,52	**94,90**	79,92	91,99	**92,25**	85,58	93,24	**93,56**
	Variome	64,37	61,36	**69,14**	84,85	81,82	**84,85**	73,20	70,13	**76,19**

Tabela 3.3: *Precision*, *recall* e *f1-score* após *fine-tuning* (retirado de [37]).

A Tabela 3.3 compara o desempenho dos modelos ao longo dos 34 GSC considerados. A coluna *'No'* refere-se ao modelo base, sem pré-treino, e as colunas *'GSPT'* e *'SSPT'* a modelos pré-treinados com recurso a 1 dos 5 GSC e SSC, respetivamente.

3.4 Modelos Linguísticos

Também eles modelos de *machine learning*, os modelos linguísticos são modelos pré-treinados especialmente desenhados para tarefas de Processamento de Linguagem Natural (PLN).

Para a tarefa de NER, dos mesmo autores de [37], em [38] é apresentado um modelo linguístico com tokenização por carateres - Hunflair, treinado ao longo de 24 milhões de *abstracts* do domínio biomédico e 3 milhões de textos completos, utilizado para a extração de 5 entidades biomédicas: *Cell Lines*, *Chemicals*, *Diseases*, *Genes* e *Species*. A avaliação ocorre num contexto de *cross-corpus*, avaliado em 3 GSC: CRAFT [3], BioNLP13 Cancer Genetics [29] e PDR. Na Tabela 3.4 o Hunflair é comparado com outras ferramentas de NLP. 'Ch' refere-se a *Chemical*, 'G' a *Gene*, 'S' a *Species* e 'D' a *'Disease*.

	CRAFT			BioNLP CG				PDR
	Ch	G	S	Ch	D	G	S	D
Misc	42,88	64,93	81,15	72,15	55,64	68,97	**80,53**	80,63
SciSpacy	35,73	47,76	54,21	58,43	56,48	66,18	57,11	75,90
HUNER	42,99	50,77	84,45	67,37	55,32	71,22	67,84	73,64
HunFlair	**59,69**	**72,19**	**85,05**	**81,82**	**65,07**	**87,71**	76,47	**83,44**

Tabela 3.4: Comparação do *f1-score* entre várias ferramentas de NER num contexto *cross-corpus*.

Já na Tabela 3.5 é feita uma comparação com aquilo que é considerado pelos autores como protótipos de investigação, isto é, abordagens de estado-de-arte desenvolvidas e publicadas mas que não se encontram prontas a usar. Também num contexto *cross-corpus* são tidos como objeto de avaliação os seguintes datasets: JNLPBA [5], BC5CDR [21] e NCBI [8].

	JNLPNA (Gene)	BC5CDR	NCBI
SciBERT	77,28	90,01	88,57
BioBERT v1.1	77,49	89,76	**89,71**
CollaboNET	**78,58**	87,68	88,60
SciSpacy	-	83,92	81,56
HunFlair	77,60	89,65	88,65
HunFlair (vanilla)	77,78	**90,57**	87,47

Tabela 3.5: Comparação do *f1-score* entre várias protótipos de investigação de NER num contexto *cross-corpus*.

Para além dos bons resultados apresentados pelo modelo proposto, realçam-se também os bons resultados de todos os outros modelos, com destaque para o SciBERT e BioBERT, dois modelos linguísticos derivados do BERT.

3.5 Sumário

Após analisadas 4 metodologias diferentes de reconhecimentos de entidades diferenciam-se dois tipos de abordagens: as que requerem a intervenção humana, nomeadamente de especialistas do domínio e as que não. Considerando que o trabalho passa por automatizar a elaboração de relatórios toxicológicos e a constante evolução da área, a primeira tipologia de abordagens é excluída à partida de se r utilizada no desenvolvimento deste trabalho, já que requereria tanto para os modelos baseados em regras 3.1 como para os modelos baseados em dicionários 3.2 um constante acompanhamento de um especialista do domínio. Já as abordagens de *Machine Learning* 3.3 e de Modelos Linguísticos 3.4 são ambas abordagens cuja aprendizagem é automática, sendo portanto a opção considerada para a anotação de entidades neste trabalho, com destaque nos modelos linguísticos cujo pré-treino facilita a tarefa de *fine-tuning*.

Capítulo 4

Abordagem

Começando pela introdução à abordagem na Secção 4.1, este capítulo foca-se na exposição detalhada de toda esta. De seguida, a Secção 4.2 descreve a arquitetura da abordagem utilizada, encerrando-se o capítulo com a Secção 4.3 onde são abordados o conjunto de atributos gerados, utilizados na identificação de frases relevantes.

Considera-se uma frase relevante toda a frase potencialmente utilizável, seja por transcrição ou adaptação, na elaboração de um relatório toxicológico.

4.1 Introdução

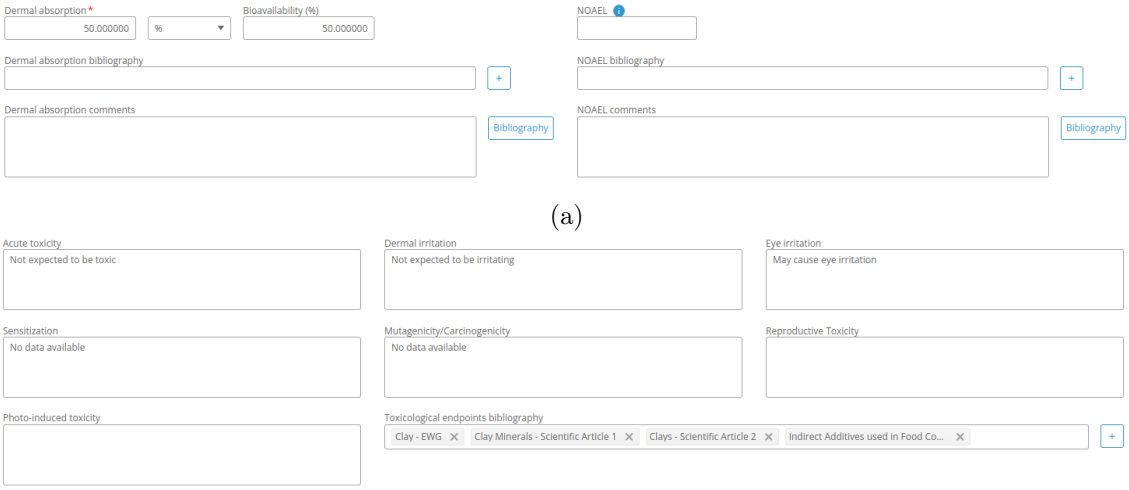
Nesta secção é feita uma introdução à abordagem passando pela descrição dos elementos químicos, documentos, anotações dos documentos e ainda dos dados de treino utilizados no *fine-tuning* dos modelos de reconhecimentos de entidades. De maneira a entender-se melhor o enquadramento não só destes dados como também da dissertação em si, a secção inicia-se com uma breve apresentação da estrutura de um relatório toxicológico.

Relembremos apenas antes de entrar em cada um destes temas e no resto do capítulo o objetivo deste trabalho: minimizar o esforço, representado pelo tempo, do avaliador de segurança na elaboração de relatórios toxicológicos de substâncias químicas - elementos, com base em documentos não estruturados.

Define-se um documento não estruturado como um documento no qual não é possível estabelecer uma associação entre as suas secções e os campos do relatório toxicológico.

4.1.1 Estrutura de um Relatório Toxicológico

O avaliador de segurança na elaboração de relatórios toxicológicos consulta diversos documentos não estruturados, elaborando relatórios toxicológicos a partir da informação recolhida destes. De modo a assegurar-se uma uniformização entre relatórios de diferentes elementos todos os relatórios obedecem à mesma estrutura. As Figuras 4.1a e 4.1b apresentam os campos de um relatório toxicológico.



(a)

(b)

Figura 4.1: Campos de um relatório toxicológico.

O avaliador de segurança tem portanto como tarefa popular os campos dos relatórios toxicológicos com informação proveniente de relatórios e artigos científicos. Cada relatório diz respeito a uma substância diferente sendo que para cada substância é necessária a consulta de vários relatórios e artigos científicos.

4.1.2 Elementos Químicos

A lista em baixo enumera os cinco elementos utilizados como referência no desenvolvimento deste trabalho, assim como o número de documentos associados a cada um, entre parêntesis, a seguir ao nome do elemento.

- Argila (2)
- Glicolípidos (2)
- Extracto de Própolis (3)
- Resveratrol (3)
- Rutin (3)

4.1.3 Documentos

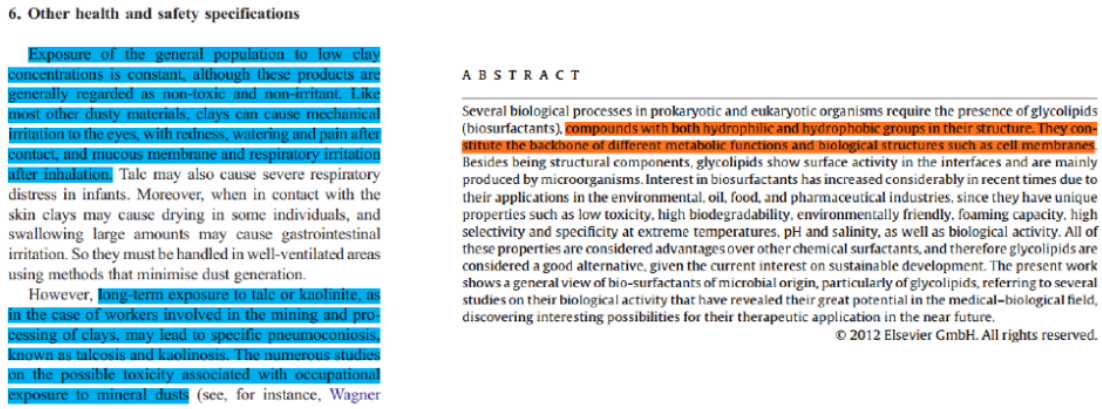
Um documento seja ele relatório ou artigo científico é sempre um ficheiro no formato pdf, utilizado pelo do avaliador de segurança como fonte de informação para a elaboração dos relatórios toxicológicos. Ao longo dos cinco elementos são considerados um total de 13 documentos, com sensivelmente 10000 palavras cada em média. A Tabela 4.1 detalha para cada documento de cada elemento o numero de páginas, frases, palavras e tempo de leitura estimado a partir de uma velocidade de leitura de 149 palavras por minuto. Note-se que o número de frases deve ser visto apenas como uma aproximação já que os valores apresentados são o resultado da transformação do ficheiro pdf num ficheiro de texto, Secção 4.2.4, sendo calculados de forma automática.

Elemento	# Documentos	# Frases / Documento	# Palavras / Documento	Minutos de Leitura / Documento
Argila	2	769	9113	∼61
		1887	20571	∼138
Glicolípidos	2	812	10591	∼71
		643	4314	∼29
Extracto de Própolis	3	1655	21443	∼144
		713	6999	∼47
		305	3184	∼21
Resveratrol	3	343	5430	∼36
		884	13158	∼88
		673	8709	∼58
Rutin	3	345	4724	∼32
		424	6573	∼44
		1455	16548	∼111

Tabela 4.1: Informação base dos documentos utilizados na elaboração de relatórios toxicológicos.

4.1.4 Anotações

Visível na Figura 4.2, no seu formato original, as anotações estão presentes nos próprios documentos por intermédio de um sublinhado e representam o conjunto de frases utilizadas na elaboração dos relatórios toxicológicos.



6. Other health and safety specifications

Exposure of the general population to low clay concentrations is constant, although these products are generally regarded as non-toxic and non-irritant. Like most other dusty materials, clays can cause mechanical irritation to the eyes, with redness, watering and pain after contact, and mucous membrane and respiratory irritation after inhalation. Talc may also cause severe respiratory distress in infants. Moreover, when in contact with the skin clays may cause drying in some individuals, and swallowing large amounts may cause gastrointestinal irritation. So they must be handled in well-ventilated areas using methods that minimise dust generation.

However, long-term exposure to talc or kaolinite, as in the case of workers involved in the mining and processing of clays, may lead to specific pneumoconiosis, known as talcosis and kaolinosis. The numerous studies on the possible toxicity associated with occupational exposure to mineral dusts (see, for instance, Wagner

ABSTRACT

Several biological processes in prokaryotic and eukaryotic organisms require the presence of glycolipids (biosurfactants), compounds with both hydrophilic and hydrophobic groups in their structure. They constitute the backbone of different metabolic functions and biological structures such as cell membranes. Besides being structural components, glycolipids show surface activity in the interfaces and are mainly produced by microorganisms. Interest in biosurfactants has increased considerably in recent times due to their applications in the environmental, oil, food, and pharmaceutical industries, since they have unique properties such as low toxicity, high biodegradability, environmentally friendly, foaming capacity, high selectivity and specificity at extreme temperatures, pH and salinity, as well as biological activity. All of these properties are considered advantages over other chemical surfactants, and therefore glycolipids are considered a good alternative, given the current interest on sustainable development. The present work shows a general view of bio-surfactants of microbial origin, particularly of glycolipids, referring to several studies on their biological activity that have revealed their great potential in the medical–biological field, discovering interesting possibilities for their therapeutic application in the near future.

Figura 4.2: Exemplo de anotações de documentos.

Atendendo ao formato em que as anotações se encontram, através de uma transcrição manual, é criado para cada documento um ficheiro de anotações. A Tabela 4.2 estabelece uma comparação entre os documentos e as suas anotações.

Elemento	# Documentos	# Anotações / Documento	# Palavras / Documento	# Palavras de Anotações / Documento	% do Documento Relevante
Argila	2	13	9113	604	6,6 %
		2	20571	86	0,4 %
Glicolípidos	2	4	10591	89	0,8 %
		3	4314	78	1,8 %
Extracto de Própolis	3	1	21443	26	0,1 %
		1	6999	33	0,5 %
		1	3184	90	2,8 %
Resveratrol	3	3	5430	83	1,5 %
		11	13158	358	2,7 %
		3	8709	65	0,7 %
Rutin	3	4	4724	71	1,5 %
		2	6573	171	2,6 %
		5	16548	71	0,4 %

Tabela 4.2: Estatísticas entre os documentos e as suas anotações.

Por análise da Tabela 4.2 conclui-se que apenas uma pequena parte de cada documento é utilizada na elaboração dos relatórios toxicológicos, corroborando a motivação base deste trabalho - minimizar o campo de procura do avaliador de segurança.

4.1.5 Entidades

Partindo do pressuposto que uma entidade é indicativa de uma frase relevante à elaboração de um relatório toxicológico surge a tarefa de reconhecimento de entidades. Define-se como entidade qualquer palavra referente a uma categoria de substantivos próprios. São consideradas três entidades distintas:

- Químicas: *bentonity*, *palygorskite*, *kaolinite*, *talc* e *sepiolite*
- Doenças: *toxicity*, *cancer*
- Espécies: *human*, *infants*

De modo a conseguir-se classificar cada palavra do documento como entidade tem-se como abordagem a utilização de modelos de reconhecimento de entidades. Considerando o levantamento feito em 2.3 é escolhido o BioBERT 2.3.2 como modelo pré-treinado, a partir do qual é feito o *fine-tuning* para as tarefas de reconhecimento de entidades. Através da biblioteca HuggingFace, uma biblioteca *open-source* de Processamento de Linguagem Natural (PLN) que coloca à disposição modelos pré-treinado como o BioBERT, são escolhidos os *datasets* apresentados na Tabela 4.3 para o treino dos modelos de reconhecimentos de entidades. Os *datasets* são escolhidos de acordo com os três tipos de entidades que se pretende classificar.

Nome	Entidades	URL
bc4chemd	Químicas	hiperligação
BC5CDR	Químicas e Doenças	hiperligação
linnaeus	Espécies	hiperligação

Tabela 4.3: *Datasets* e as suas entidades (consultado a 25/07/2022).

A partir destes dados são treinados três modelos: um de reconhecimento de entidades químicas, outro de reconhecimento de entidades químicas e doenças e outro de reconhecimento de entidades de espécies.

4.2 Arquitetura

Nesta secção é apresentada a arquitetura proposta, a estrutura dos ficheiros e a arquitetura do programa utilizado na geração e avaliação dos atributos. A secção começa com uma visão global da arquitetura, entrando de seguida, de forma sequencial, em detalhe de cada uma das suas componentes, incluindo a componente responsável pela avaliação dos atributos gerados. 4.2.7.

4.2.1 Proposta de Arquitetura

Representado na Figura 4.3 encontra-se a proposta de arquitetura para a realização deste trabalho. A arquitetura conta com 8 módulos, um dos quais para a avaliação, responsáveis por todo o processo desde a receção e tratamento de dados até ao *output* a ser utilizado pela plataforma CosmeDesk

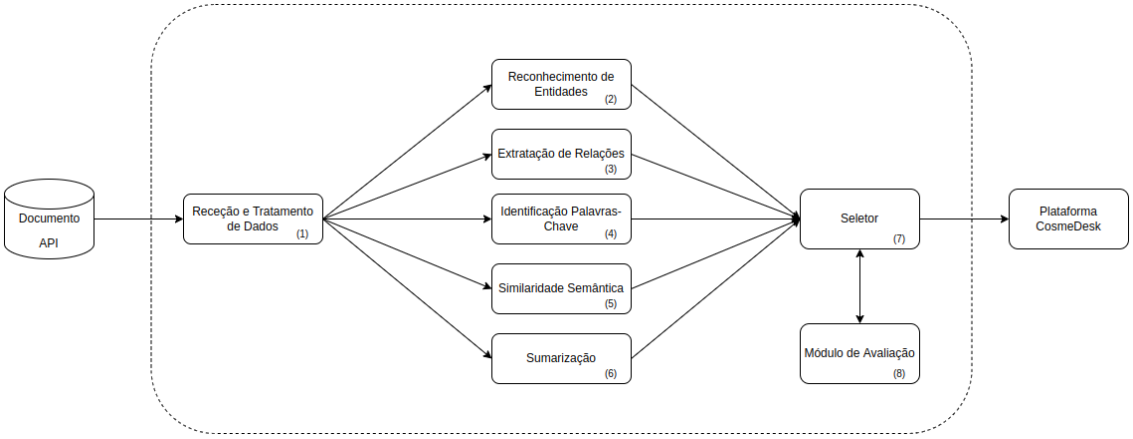


Figura 4.3: Arquitetura proposta.

4.2.2 Visão Geral

Seguindo os moldes da arquitetura proposta 4.2.1, a Figura 4.4 ilustra graficamente as 4 componentes pelo que o programa é composto:

- PdfExtractor: Conversão do ficheiro pdf num ficheiro de texto - (1).
- Summarizer: Sumarização do ficheiro de texto através de sumarização extrativa - (6).
- CosmeDesk: Geração de atributos associadas a cada frase dos ficheiros de texto, sumarizados ou não - (2), (4), (5) e (7).
- Metric: Avaliação das frases consideradas relevantes - (8).

O programa parte do pressuposto que os os elementos e os documentos a serem analisados se encontram em conformidade com a estrutura descrita em 4.2.3

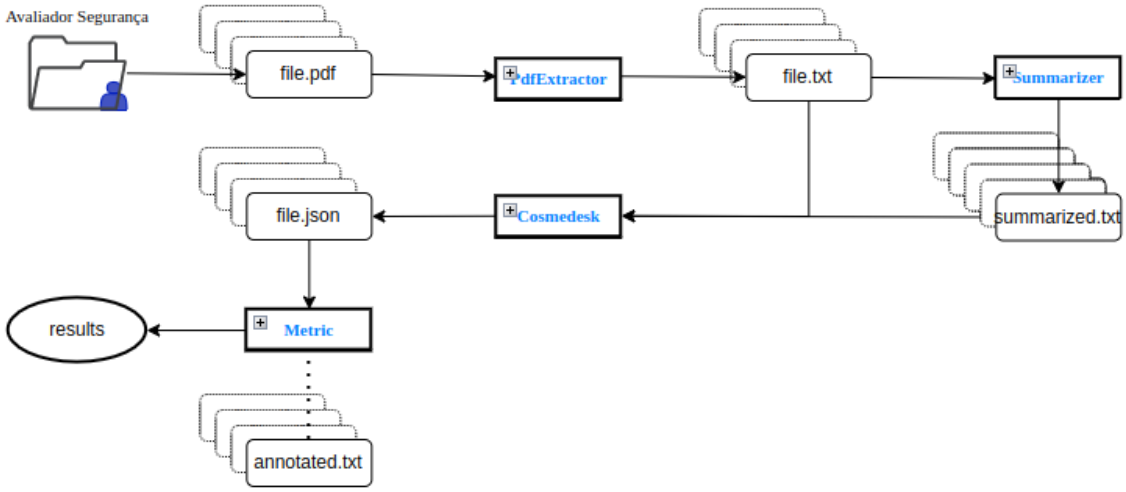


Figura 4.4: Visão geral da arquitetura do programa.

4.2.3 Estrutura dos Ficheiros

De maneira a que o programa funcione corretamente é necessária a existência de uma estrutura dos ficheiros a serem utilizados e gerados pelo programa, representada na Figura 4.5.

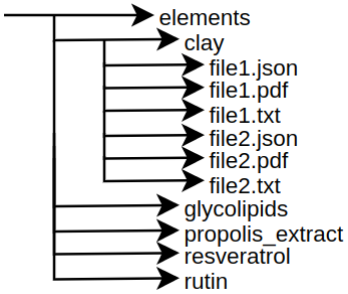


Figura 4.5: Estrutura dos ficheiros.

Todos os ficheiros utilizados e gerados encontram-se dentro de uma pasta apelidada de *elements*. A pasta organiza-se de modo a que cada elemento seja também ele uma pasta dentro desta. Dentro das pastas dos elementos encontram-se todos os ficheiros utilizados. Os documentos originais (.pdf) , os ficheiros de texto (.txt), resultado da conversão dos ficheiros .pdf em ficheiro de texto e, por fim, os ficheiros finais (.json) relativos aos atributos gerados, associados a cada frase de um documento. Os nome dos ficheiros gerados pelo programa permanecem iguais aos documentos originais alterando-se apenas a extensão do ficheiro.

4.2.4 PdfExtractor

O PdfExtractor é a componente responsável pela conversão dos documentos originais em ficheiros de texto. A Figura 4.6 apresenta a representação de alguns atributos e métodos da classe responsável pela implementação da componente.

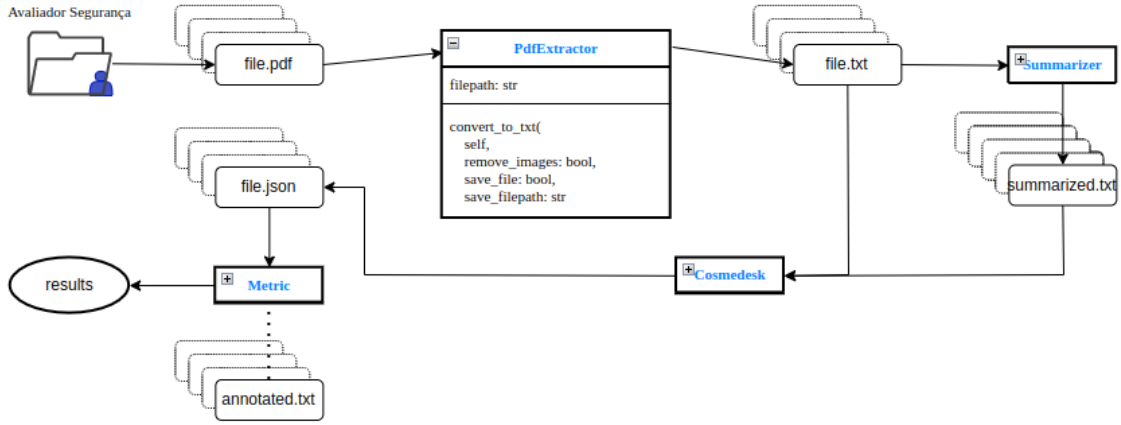


Figura 4.6: Arquitetura do programa - PdfExtractor.

Para além do método visível na Figura 4.6, que faz a conversão de um documento num ficheiro de texto, resultando em blocos extensos, compostos por várias frases, é também feita a segmentação dos blocos originais em frases, ou seja, em blocos menores. Recorrendo à biblioteca NLTK, um *toolkit* de linguagem natural, a componente internamente implementa um algoritmo de tokenização dos blocos em frases. É tido como um exemplo deste processo a Figura 4.7 que representa o resultado da divisão de um bloco em frases.

A B S T R A C T

Clays and clay minerals are widely used in many facets of our society. This review addresses the main clays of each phyllosilicate groups, namely, kaolinite, montmorillonite (Mt) and sepiolite, placing special emphasis on Mt and kaolinite, which are the clays that are more frequently used in food packaging, one of the applications that are currently exhibiting higher development. The improvements in the composite materials obtained from clays and polymeric matrices are remarkable and well known, but the potential toxicological effects of unmodified or modified clay minerals and derived nanocomposites are currently being investigated with increased interest. In this sense, this work focused on a review of the published reports related to the analysis of the toxicological profile of commercial and novel modified clays and derived nanocomposites. An exhaustive review of the main *in vitro* and *in vivo* toxicological studies, antimicrobial activity assessments, and the human and environmental impacts of clays and derived nanocomposites was performed. From the analysis of the scientific literature different conclusions can be derived. Thus, *in vitro* studies suggest that clays in general induce cytotoxicity (with dependence on the clay, concentration, experimental system, *etc.*) with different underlying mechanisms such as necrosis/apoptosis, oxidative stress or genotoxicity. However, most of *in vivo* experiments performed in rodents showed no clear evidences of systemic toxicity even at doses of 5000 mg/kg. Regarding to humans, pulmonary exposure is the most frequent, and although clays are usually mixed with other minerals, they have been reported to induce pneumoconiosis *per se*. Oral exposure is also common both intentionally and unintentionally. Although they do not show a high toxicity through this pathway, toxic effects could be induced due to the increased or reduced exposure to mineral elements. Finally, there are few studies about the effects of clay minerals on wildlife, with laboratory trials showing contradictory outcomes. Clay minerals have different applications in the environment, thus with a strict control of the concentrations used, they can provide beneficial uses.

Figura 4.7: Segmentação de um bloco em frases.

Apesar de situações como "*...induce cytotoxicity (with dependence on the clay, concentration, experimental system, etc.)* || *with different underlying mechanisms...*"onde o processo da segmentação divide uma frase a meio, considera-se o benefício superior à perda. Desta maneira passa a ser possível identificar apenas alguns excertos como relevantes ao invés da totalidade do parágrafo sendo que, caso a venha a ser necessário, não deixa de existir a possibilidade de marcar o parágrafo inteiro como relevante sob a condição de haver um ou mais excertos identificados como tal.

4.2.5 Summarizer

Ilustrado na Figura 4.8 o Summarizer é a componente responsável pela sumarização do texto, isto é, a redução do texto às ideias principais. A sumarização é alcançada através de 4 métodos extrativos: LexRank, Luhn, Lsa e TextRank. Por constrangimentos da metodologia de avaliação descrita em 4.2.7 e 5.1 não foram utilizados métodos de sumarização abstrativa.

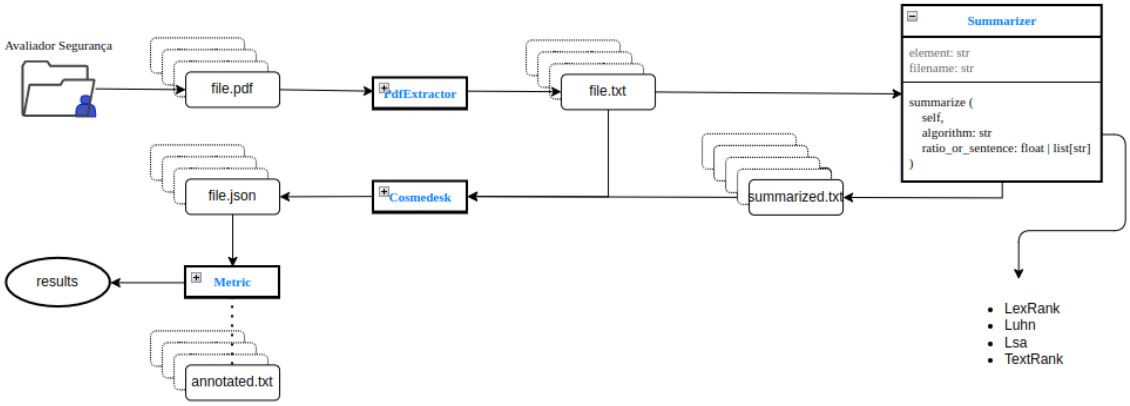


Figura 4.8: Arquitetura do programa - Summarizer.

Conforme visível na Figura 4.8 o Summarizer no seu método de sumarização recebe dois argumentos: *algorithm* e *ratio_or_sentence*. O primeiro argumento diz respeito a um dos quatro algoritmo mencionados anteriormente e o segundo argumento diz respeito ao tamanho do documento sumariado, dado ou por uma percentagem relativa ao documento ou pelo número de frases pretendido. Desta componente resulta uma versão sumarizada do ficheiro original de acordo com o algoritmo e rácio de compressão escolhidos.

4.2.6 CosmeDesk

Composto atualmente por duas sub-componentes o CosmeDesk, representado na Figura 4.9, apresenta-se como a componente principal do programa. É no CosmeDesk que os atributos entidades, palavras-chave e similaridade semântica, utilizados na identificação de frases relevantes dos documentos.

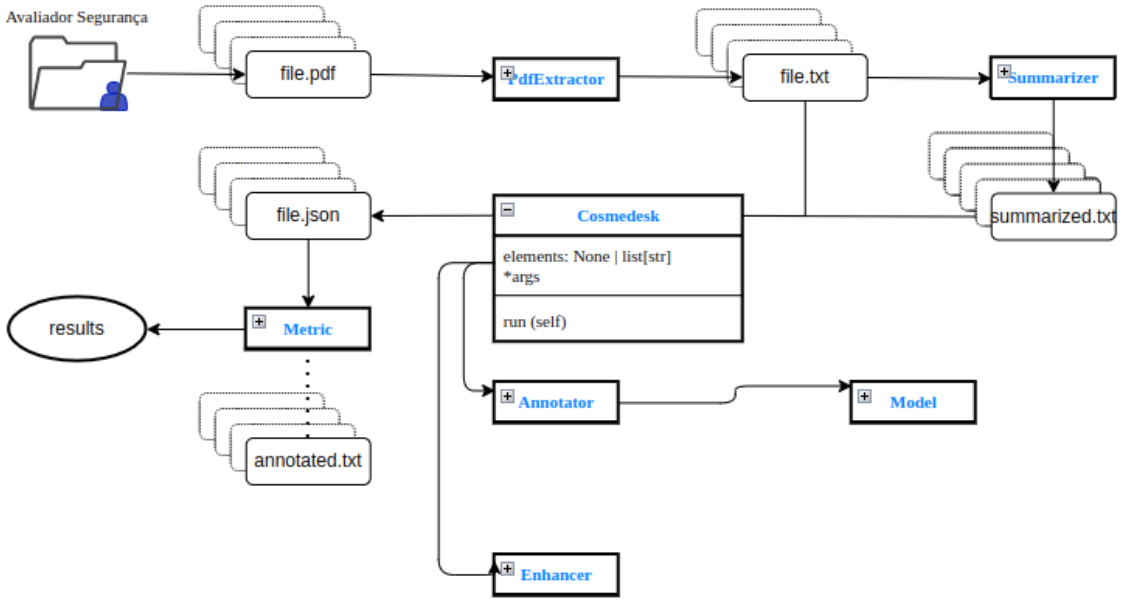


Figura 4.9: Arquitetura do programa - CosmeDesk.

Conforme visível na Figura 4.9 a componente recebe uma lista de elementos e um número indeterminado de argumentos. A lista de elementos determina os elementos a serem iterados pelo programa e a lista de argumentos recebe as sub-componentes geradoras dos atributos com base nos quais as frases são classificadas como relevantes. Para as sub-componentes poderem ser utilizadas na componente CosmeDesk têm de respeitar duas condições:

1. Ter um método run(self, sentences). Este método deverá ser o método principal, sendo as sentences uma lista de strings referentes às várias frases de um documento.

2. Retornar no método run() um dicionário com os atributos gerados. O dicionário deverá ter como chaves os nomes nomes dos atributos e como valores das chaves uma lista com os atributos gerados para cada frase.

A maneira como o CosmeDesk está construído origina um programa versátil possibilitando a adição e remoção de sub-componentes, tornando-o numa ferramenta tanto exploratória como de produção. A utilização em código do CosmeDesk juntamente das duas sub-componentes encontra-se visível na Figura 4.10.

main.py

```
# None - Iterates over all elements
elements = None

annotator = Annotator(models)
enhancer = Enhancer(keywords=True, semantic_similarity=True)

cosmedesk = Cosmedesk(
    elements,
    annotator,
    enhancer
)

cosmedesk.run()
```

Figura 4.10: Exemplo de código do funcionamento do programa.

Como representado na Figura 4.10 o CosmeDesk recebe uma lista de elementos que, quando a None, itera sobre todos os elementos definidos na estrutura dos ficheiros (Secção 4.2.3). Os restantes argumentos são sub-componentes, objetos de classes, que respeitam as condições definidas anteriormente. Ao correr o CosmeDesk temos como resultado um ficheiro .json composto pelos atributos implementados pelas sub-componentes.

São agora detalhadas as duas sub-componentes utilizadas no CosmeDesk. O Annotator, responsável pela anotação de entidades e o Enchancer, responsável pela identificação de palavras-chave e similaridade semântica entre frases.

Annotator

O Annotator, Figura 4.11, é responsável pela anotação de entidades e identificação das frases do documento que as contêm. Tem como atributos uma lista de Models, Figura 4.12, e retorna um dicionário onde para cada frase de um documento indica se a frase contem entidades, identifica-as e apresenta ainda algumas estatísticas relativas às entidades reconhecidas.

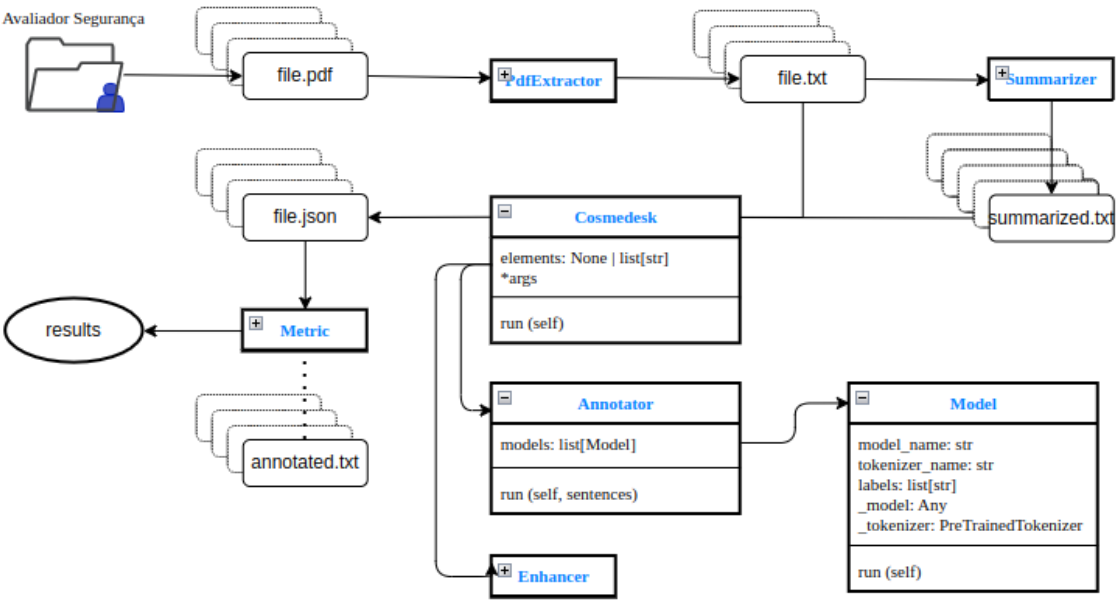


Figura 4.11: Arquitetura do programa - Annotator.

Indo de encontro à Secção 4.1.5 onde são detalhadas as entidades, os dados de treino e os modelos resultantes, todos os modelos utilizados no Annotator são modelo pré-treinados com *fine-tuning* para a tarefa de reconhecimento de entidades. Estes modelos encontram-se guardados numa diretoria do programa e são carregados no Annotator através do ficheiro de configuração da Figura 4.12. Analogamente ao CosmeDesk o Annotator é versátil nos modelos que podem ser utilizados bastando adicionar qualquer modelo pré-treinado na diretoria respetiva e adicionar as suas informações ao ficheiro de configuração.

models configs

```
models = [
    {
        'model_name': 'biobert_bc4chemd',
        'tokenizer_name': 'dmis-lab/biobert-base-cased-v1.1',
        'labels': ['O', 'B-C', 'I-C']
    },
    {
        'model_name': 'biobert_bc5cdr_chemical_disease',
        'tokenizer_name': 'dmis-lab/biobert-base-cased-v1.1',
        'labels': ['O', 'B-D', 'I-D', 'B-C', 'I-C']

    },
    {
        'model_name': 'biobert_linnaeus',
        'tokenizer_name': 'dmis-lab/biobert-base-cased-v1.1',
        'labels': ['O', 'B-S', 'I-S']
    }
]
```

Figura 4.12: Ficheiro de configuração - Models.

A Figura 4.12 apresenta o ficheiro de configuração para os três modelos utilizados. Cada modelo é caracterizado pelo seu nome, pelo nome do *tokenizer* e as entidades a

serem anotadas, ou seja, as *labels*. A identificação de entidades é realizada segundo a abordagem de BIO tagging 2.2.1. Consideram-se ao longo dos três modelos as seguintes *labels*:

- O: Entidade não associada
- B-C: Início de uma entidade química *(Chemical)*
- I-C: Continuação de uma entidade química *(Chemical)*
- B-D: Início de uma entidade de doenças *(Diseasse)*
- I-D: Continuação de uma entidade de doenças *(Diseases)*
- B-S: Início de uma entidade de espécies *(Species)*
- I-S: Continuação de uma entidade de espécies *(Species)*

Enhancer

O Enhancer, Figura 4.13, é responsável pela identificação de palavras-chave e pelo calculo da similaridade semântica entre frases de um documento e um conjunto pré-definido de expressões. Seguindo a lógica do resto do programa, tanto as palavras-chave como as expressões utilizadas no cálculo da similaridade semântica são também configuráveis através de um ficheiro de configuração. De seguida são apresentadas as palavras-chave e as expressões utilizadas.

Palavras-chave :

- *Sensitization*
- *Mutagenicity*
- *Carcinogenicity*
- *Irritation*
- *Toxicity*
- *Dermal*
- *Eye*
- *Acute*
- *Reproductive*
- *Photo-induced*

Expressões :

- *No observed adversed effect level*
- *Sensitization*
- *Mutagenicity/Carcinogenicity*
- *Dermal irritation*
- *Eye irritation*
- *Acute toxicity*
- *Reproductive toxicity*
- *Photo-induced toxicity*

Tanto as palavras-chave como as expressões derivam de campos do relatório toxicológico (figuras 4.1a e 4.1b).

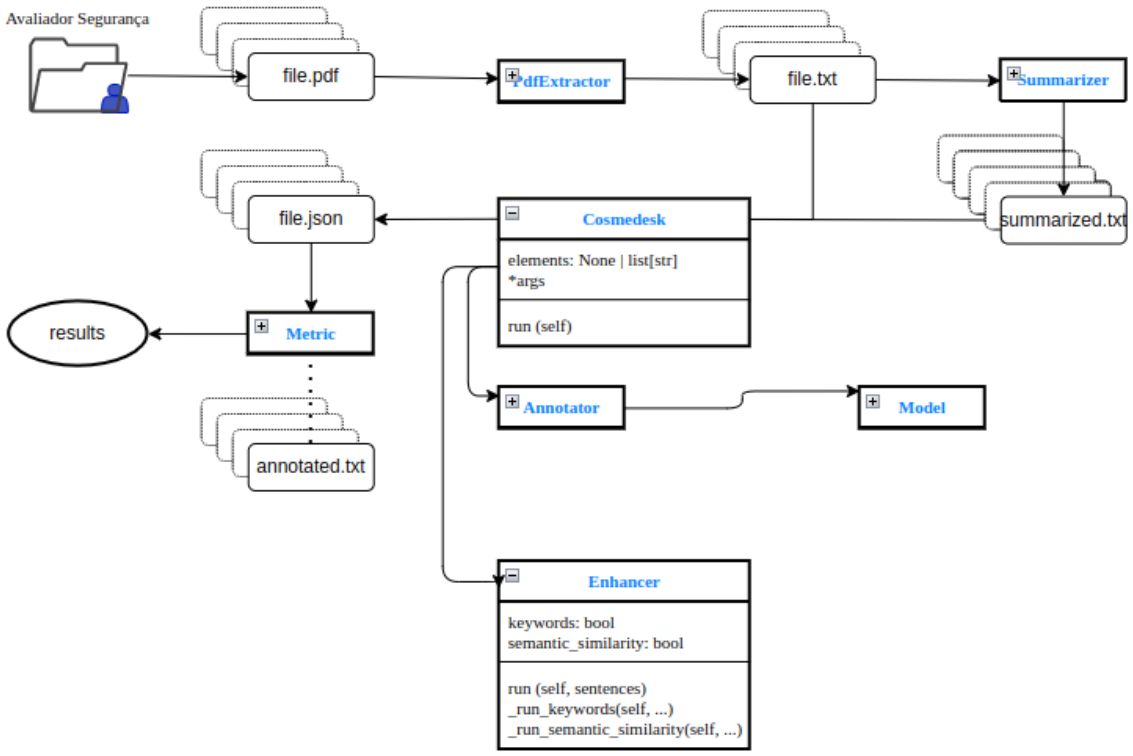


Figura 4.13: Arquitetura do programa - Enhancer.

Conforme visível na Figura 4.2.6 o Enhancer é constituído por dois métodos, sendo um para detetar se uma frase contem palavras chave e outro para o cálculo da similaridade semântica para com as expressões apresentadas acima. O Enhancer recebe como atributos uma *flag* para cada um dos métodos implementados indicativa de se queremos utilizá-los ou não.

4.2.7 Metric

A Metric, Figura 4.14, é a componente responsável pela análise e validação dos atributos gerados. Esta componente permite avaliar não só cada atributo individualmente como também combinações dos vários atributos. A avaliação é feita de forma automática tendo como referência as anotações referidas em 4.1.4.

Para efeitos de avaliação são considerados três parâmetros de avaliação:

- Verdadeiros Positivos: Frases classificadas como relevantes efetivamente relevantes, isto é, utilizadas na elaboração de um relatório toxicológico.
- Falsos Positivos: Frases classificadas como não relevantes, relevantes.
- Ganho Textual (%): Percentagem de redução do texto face ao documento original.

A explicação dos parâmetros de avaliação utilizados é feita na Secção 5.1.1

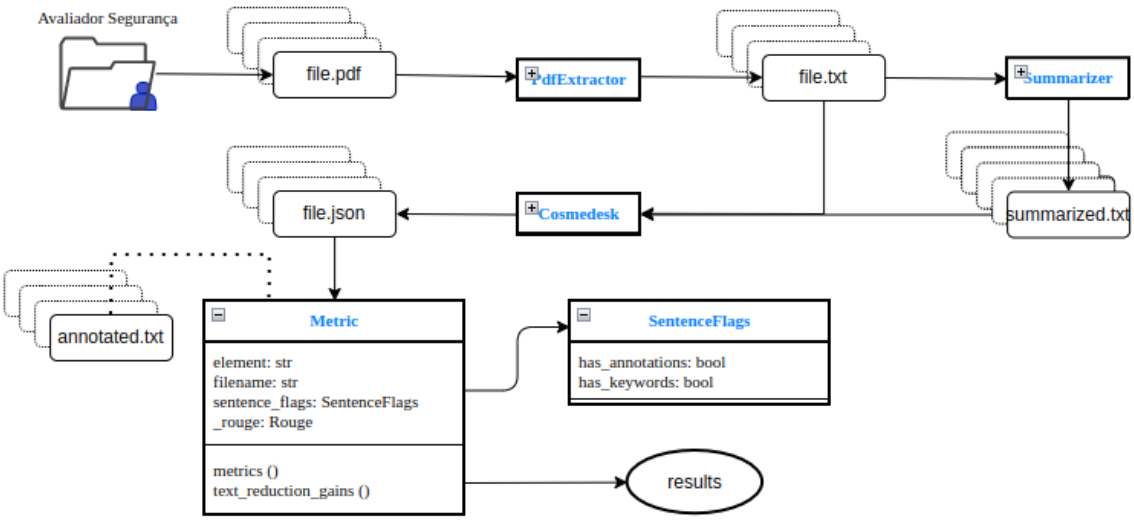


Figura 4.14: Arquitetura do programa - Metric.

Na Figura 4.14, verificamos que a Metric tem como atributo *sentence_flags*, onde é feita a seleção dos atributos ou combinações de atributos a avaliar, e o *rouge*, um algoritmo que mede a similaridade entre frases utilizado no mapeamento entre as frases consideradas relevantes e as anotações. Tanto o *rouge* como as *sentence_flags* são desenvolvidos em 5.1.

4.2.8 DocumentFetcher

Por fim, apesar de não estar inserido na pipeline do programa e surgindo como uma alternativa à importação manual dos documentos, temos o DocumentFetcher. Tendo como entrada um termo de pesquisa, a componente é responsável por apresentar ao utilizador uma lista de documentos relacionados onde, através do título e resumo, o avaliador de segurança escolhe os documentos a serem carregados para o programa.

4.3 Atributos

Nesta secção são descritos os atributos gerados pelo programa através dos quais são selecionadas as frases relevantes de um documento. Relembrando a estrutura descrita em 1.5, os atributos são guardados num ficheiro com o mesmo nome do documento original, diferenciando-se apenas do mesmo pela extensão do ficheiro. O ficheiro gerado com os atributos é um ficheiro .json contendo para cada frase do documento em análise os atributos associados. Apesar de serem gerados mais de três atributos, para efeitos práticos consideram-se apenas os seguintes três:

- *has_annotations*: Indica se uma frase contem anotações.
- *has_keywords*: Indica se uma frase contem palavras-chave.
- *semantic_similarity*: Apresenta a similaridade semântica entre uma frase e um conjunto pré-definido de expressões.

Retirado de um ficheiro .json, a Figura 4.15 mostra todos os atributos associados a uma frase. Os atributos *annotations_stats*, *annotations* e *semantic_similarity* são proposita-

damente deixados em branco de maneira a não introduzir informação excessiva na figura, sendo abordados nas secções seguintes.

file struct

```
{
  "block_nr": 51,
  "sentence": "Of all the phyllosilicates, only some clay minerals are used in pharmacy and
cosmetics, including kaolinite, talc, smectites and fibrous clays.",
  "has_annotations": 1,
  "annotations_stats": {},
  "annotations": [],
  "has_keywords": 0,
  "semantic_similarity": {}
}
```

Figura 4.15: Atributos associados a uma frase.

Conforme referido, existem mais que três atributos nomeadamente *annotations_ stats* e *annotations*, sendo através destes atributos que o atributo *has_ annotations* 4.3.1 é criado. Para além dos atributos cada instância do ficheiro .json é caracterizada por uma frase (*sentence*) e pelo número da frase (*block_ nr*) a quais os atributos pertencem.

4.3.1 *has_ annotations*

As anotações são criadas através do Annotator 4.2.6 e indicam-nos se uma frase contem presente alguma entidade. Assim, o atributo *has_ annotations* tem dois valores possíveis: 1, caso a frase tenha uma ou mais entidades anotadas e 0, caso não existam anotações. O atributo *has_ annotations* deriva de dois outros atributos, o *annotations_ stats* e o *annotations*.

annotations_ stats

O *annotations_ stats* dá-nos algumas estatísticas das entidades anotadas. As entidades anotadas referem-se às entidades anotadas pelos modelos em uso conforme descrito em 4.1.5 e 4.2.6.

annotations_stats

```
"annotations_stats": {
  "O": 105,
  "B-C": 7,
  "I-C": 17,
  "B-D": 0,
  "I-D": 0,
  "B-S": 0,
  "I-S": 0
}
```

Figura 4.16: Atributos associados a uma frase - *annotations_ stats*.

Olhando para a Figura 4.16 que representa o atributo das *annotations_ stats* da Figura 4.15, retiramos que para a frase em questão foram detetadas sete entidades químicas, não tendo sido detetadas nenhuma entidade de doenças ou espécies.

annotations

Para cada modelo em uso, as *annotations* dão-nos todas as anotações associadas aos tokens de uma frase. Nas figuras 4.17a e 4.17b é apresentado o atributo *annotations* para o modelo biobert_bc4chemd.

(a) (b)

Figura 4.17: Atributos associados a uma frase - *annotations*.

De acordo com as figuras 4.17a e 4.17b verificamos que o atributo é composto pelo nome do modelo, as labels associadas ao modelo, estatísticas num formato semelhante a *annotations_ stats* com a difença de serem apenas respetivas ao modelo em questão e não a todos, e uma lista de tokens da frase correspondente assim como outra lista com as anotações associadas a cada token pelo índice da lista.

4.3.2 *has_ keywords*

As palavras-chave, originadoras do atributo *has_ keywords* são geradas no Enhancer 4.2.6 e averiguam se uma frase contem algumas das palavras-chave presentes no ficheiro de configuração. O atributo *has_ keywords* segue a mesma lógica do atributo *has_ annotations* com dois valores possíveis: 1 caso a frase contenha uma ou mais palavras chave e 0 na situação contrária

4.3.3 *semantic_ similarity*

Por fim, a similaridade semântica gerada também no Enhancer 4.2.6 calcula a similaridade entre uma frase e um conjunto pré-definido de expressões. Para o cálculo da similaridade semântica é gerado o *embedding* da frase e o *embedding* de cada expressão com que a frase vai ser comparada. Através do *cosine score* é calculada a similaridade semântica que se

resume ao ângulo formado pelos vetores dos *embeddings*. Quanto menor o ângulo entre os dois vetores mais similares são. O valor da similaridade semântica compreende-se entre -1 e 1, com 1 a representar o máximo de similaridade. A Figura 4.18 mostra um exemplo da anotação de uma frase segundo este atributo.

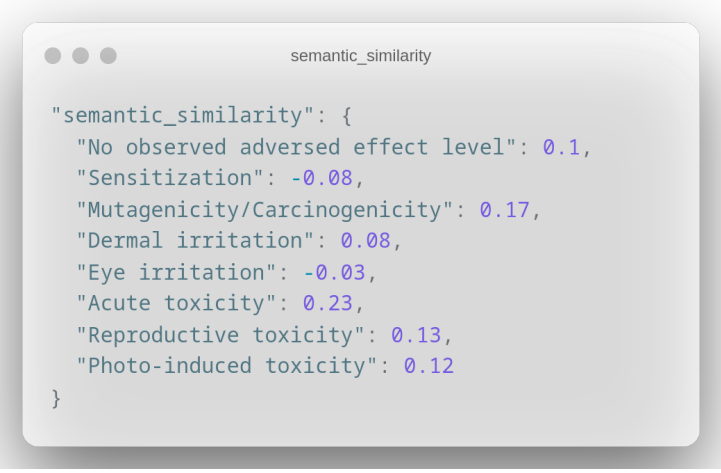


Figura 4.18: Atributos associados a uma frase - *semantic_ similarity*.

Conforme podemos verificar na Figura 4.18 as expressões são essencialmente campos do relatório toxicológico a preencher. Para cada expressão existe um valor de similaridade semântico associado à frase. No caso da frase da Figura 4.15 "*Of all the phyllosilicates, only some clay minerals are used in pharmacy and cosmetics, including kaolinite, talc, smectites and fibrous clays.* a *Acute toxicity*"é a expressão que mais se aproxima da frase com similaridade semântica igual a 0,23 seguida pela *Mutagenicity/Carcinogenicity* com 0,17.

Capítulo 5

Experimentação e Resultados

O presente capítulo apresenta os resultados da abordagem utilizada, dividindo-se em duas partes. A primeira relativa à metodologia de avaliação 5.1 e a segunda relativa à análise de resultados 5.2.

5.1 Metodologia de Avaliação

Esta secção diz respeito à explicação dos parâmetros de avaliação utilizados e à metodologia de seleção das frases relevantes, incluindo os métodos implementados na componente Metric 4.14.

5.1.1 Parâmetros de Avaliação

De maneira a conseguir-mos avaliar a qualidade das frases relevantes inferidas, de acordo com cada atributo ou combinações de atributos, consideraram-se três parâmetros de avaliação: verdadeiros positivos, falsos negativos e o ganho textual. Continuando a definição dada em 4.2.7 os VPs traduzem-se em quantas das anotações foram detetadas através de frases consideradas relevantes e os FNs o contrário. Se considerarmos a totalidade das frases de um documento como relevantes, todas as as frases correspondentes às anotações serão logicamente sinalizadas, não existindo no entanto vantagem com esta abordagem pois o avaliador de segurança teria de ler o documento na integra. Assim, é estabelecido o parâmetro de ganho textual que diz respeito à redução do texto, dada por uma percentagem, face ao documento original. Através destes três parâmetros conseguimos analisar o qualidade das frases relevantes extraídas em função do número de anotações detetadas, não detetadas e do ganho textual.

5.1.2 Frases Relevantes

Conforme referido ao longo do documento o objetivo da abordagem tomada passa pela identificação de frases relevantes. Relembrando a definição de frase relevante, uma frase relevante é toda a frase potencialmente utilizável, seja por transcrição ou adaptação, na elaboração de um relatório toxicológico. Através destas podemos restringir o espaço de leitura no documento evitando ao avaliador de segurança a sua leitura na integra, onde num cenário ideal seriam apenas selecionadas pelo algoritmo as frases efetivamente necessárias à construção do relatório. Infelizmente esta tarefa não é trivial já que na maioria das

situações nem 2% (Tabela 4.2) do documento chega a ser utilizado na construção de um relatório.

Rouge Score

Através do mapeamento entre as frases de um documento e as anotações, o *rouge score* permite fazer a avaliação das frases consideradas relevantes, possibilitando a avaliação dos atributos gerados, quer individualmente quer pela sua combinação. Convém então entender o que é o *rouge score* e como nos é útil.

Na sua variante mais simples, *rouge-n*, o *rouge* mede o número de n-gramas correspondentes entre uma frase considerada relevante e uma anotação. A proximidade entre estas é dada por três métricas:

- $Recall = \frac{\text{nr de n-gramas da frase} \cap \text{nr de n-gramas da anotação}}{\text{nr de n-gramas da anotação}}$
- $Precison = \frac{\text{nr de n-gramas da frase} \cap \text{nr de n-gramas da anotação}}{\text{nr de n-gramas da frase}}$
- $F1 = 2 * \frac{\text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}}$

Como uma frase considerada relevante é uma frase anotada, completa ou parcial, utiliza-se o *rouge-l* onde em vez de considerar o nr de n-gramas da frase ∩ nr de n-gramas da anotação considera-se a maior subsequência comum. Consideremos o seguinte exemplo:

Frase: *The crystalline silica content will often be the decisive factor in clay-induced adverse health effects*

Anotação: *crystalline silica*

A maior subsequência comum é *crystalline silica* com tamanho de 2. Assim, obtemos para o exemplo um *recall* de 1, *precison* de 0.13, e *f1* de 0.24.

Considera-se existir uma correspondência entre uma frase e uma anotação quando uma das seguintes condições é respeitada:

- Recall >= 0.8
- Precision >= 0.8
- F1 >= 0.6

Para cada anotação de um documento é utilizado o *rouge* em todas as frases consideradas relevantes. O número de anotações com correspondência equivale aos VPs e o número de anotações menos o número de VPs aos FNs.

Sentence Flags

As *sentence flags* permitem-nos testar várias configurações, isto é, várias combinações de atributos. Dos três atributos gerados apenas dois, o *has_annotations* e o *has_keywords* são considerados nesta abordagem. Os estados que cada um dos dois atributos pode tomar são os seguintes:

- *has_ annotations*: [*True*, *False*, *None*]
- *has_ keywords*: [*True*, *False*, *None*]

Pelas combinações dos atributos geram-se nove configurações diferentes, utilizando-se cinco:

- *SentenceFlags(has_ annotations=True, has_ keywords=True)*
- *SentenceFlags(has_ annotations=True, has_ keywords=False)*
- *SentenceFlags(has_ annotations=True, has_ keywords=None)*
- *SentenceFlags(has_ annotations=False, has_ keywords=True)*
- ~~*SentenceFlags(has_ annotations=False, has_ keywords=False)*~~
- ~~*SentenceFlags(has_ annotations=False, has_ keywords=None)*~~
- *SentenceFlags(has_ annotations=None, has_ keywords=True)*
- ~~*SentenceFlags(has_ annotations=None, has_ keywords=False)*~~
- ~~*SentenceFlags(has_ annotations=None, has_ keywords=None)*~~

Traduzindo o significado de uma configuração ao termos *has_ annotations = True* e *has_ keywords = True* são sinalizadas como relevantes todas as frases que contêm anotações e palavras-chave, no caso de termos *has_ keywords = False* em vez de *True* são consideradas relevantes as frases com anotações e sem palavras-chave e por fim, se tivermos *has_ keywords = None*, consideram-se frases com anotações ignorando-se o atributo das palavras-chave. Assim sendo, considera-se no campo de procura das configurações apenas as configurações que considerem frases com pelo menos um atributo presente.

Similaridade Semântica

A similaridade semântica 4.3.3 é o terceiro dos três atributos gerados e permite estabelecer uma medida de similaridade entre cada frase do documento e um conjunto de expressões pré-definidas. Para cada frase, entre todas as expressões, tem-se como referencia o maior valor de similaridade. Uma frase é considerada relevante quando o valor de referencia é superior ou igual ao *threshold* definido. Tomando como exemplo a figura 4.18 o valor de 0.23 é o valor de referencia e a frase é considerada relevante caso o *threshold* definido seja igual ou inferior a este valor.

Sumarização

A sumarização 4.2.5 apesar de não gerar nenhum atributo é também ela um mecanismo de identificação de frases relevantes. Atendendo a que apenas é utilizada sumarização extrativa as frases extraídas são em si as frases consideradas relevantes. As frases extraídas dependem do algoritmo utilizado e rácio de sumarização escolhido.

5.1.3 Redução de Texto

O ganho textual é o terceiro e último parâmetro de avaliação considerado e permite-nos estabelecer uma relação custo benefício entre o número de anotações identificadas e a redução de texto associada, isto é, a quantidade de texto para cada documento que o avaliador de segurança não tem de analisar. Esta redução é calculada com base no número total de palavras de um documento e o número de palavras das frases consideradas relevantes, assim:

$$\text{Ganho Textual } (\%) = 1 - \frac{\text{nr frases consideradas relevantes}}{\text{nr frases documento}}$$

5.2 Análise de Resultados

Nesta secção serão analisados os resultados obtidos em função dos três parâmetros de avaliação escolhidos em 5.1.1. Apesar de serem todos sujeitos à mesma avaliação podemos classificar os resultados em três grupos distintos mediante a abordagem de extração de frases relevantes:

- *SentenceFlags*: Extração baseada nos atributos *has_annotations* e *has_keywords*, individualmente ou através das suas combinações.
- Similaridade Semântica: Extração baseada no atributo *semantic_similarity*.
- Sumarização: Extração baseada em métodos de sumarização extrativa.

5.2.1 *SentenceFlags*

Começando pela análise das configurações enunciadas em 5.1.2, a Tabela 5.1 apresenta os resultados relativos aos atributos *has_annotations* e *sentence_flags*.

SentenceFlags		Parâmetros Avaliação		
has_annotations	has_keywords	VPs	FNs	Ganho (%)
True	True	12	41	95
True	False	35	18	44
True	None	39	14	38
False	True	2	51	99
None	True	13	40	94

Tabela 5.1: Configurações *SentenceFlags* ao longo de todos os documentos considerados.

Conforme podemos verificar existe uma grande disparidade entre os resultados da Tabela 5.1. Tanto a configuração 1 como as configurações 4 e 5 apresentam ganhos na redução de texto muito elevados, o que pouparia ao avaliador de segurança bastante tempo na elaboração dos relatórios toxicológicos. No entanto, o número de anotações detetadas, dado pelos VPs, é reduzido e impraticável já que a maioria das frases efetivamente relevantes não são detetadas. Por outro lado, as configurações 2 e 3 apresentam valores mais elevados

de VPs a troco de uma redução acentuada no ganho. A configuração 2, com 39 anotações detetadas num total de 53 e 44% de redução textual, e a configuração 3, com mais 4 anotações detetadas a troco de uma redução no ganho. Apesar da grande disparidade entre resultados e da escolha da configuração depender da preferência pessoal do avaliador, podemos concluir que a deteção de frases relevantes depende quase exclusivamente do atributo *has_ annotations*, não sendo o atributo *has_ keywords* relevante na sua deteção.

5.2.2 Similaridade Semântica

A Tabela 5.2 apresenta os resultados referentes à similaridade sehttps://opensea.io/collection/voyagers-genesis mântica em função de diferentes *thresholds*. Para a configuração 1 são consideradas relevantes todas as frases que tenham um valor de referencia igual ou superior a 0.15. Das 53 frases efetivamente relevantes foram detetadas 47 tendo-se reduzido o tamanho do documento em 7%.

	Threshold Similaridade Semântica	Parâmetros Avaliação		
		VPs	FNs	Ganho (%)
Config 1	0,15	47	6	7
Config 2	0,2	47	6	19
Config 3	0,25	45	8	35
Config 4	0,3	38	15	57
Config 5	0,35	23	30	76
Config 6	0,4	6	47	90

Tabela 5.2: *Thresholds* similaridade semântica ao longo de todos os documentos considerados.

Novamente a seleção da configuração depende da preferência pessoal do avaliador, destacando-se no entanto as configurações 3 e 4, cujos compromissos entre o ganho e os VP's são dos mais equilibrados quando comparados com as outras configurações.

5.2.3 Sumarização

Com base em mecanismos de sumarização, na Tabela 5.3 são visíveis os resultados relativos à extração de frases relevantes através desta abordagem.

	Racio	Algoritmo	Parâmetros Avaliação		
			VPs	FNs	Ganho (%)
Config 1	0,8	LexRank	49	4	20
Config 2	0,8	Lsa	49	4	20
Config 3	0,8	Luhn	49	4	20
Config 4	0,8	TextRank	49	4	20
Config 5	0,6	LexRank	49	4	40
Config 6	0,6	Lsa	49	4	40
Config 7	0,6	Luhn	49	4	40
Config 8	0,6	TextRank	48	5	40
Config 9	0,4	LexRank	39	14	60
Config 10	0,4	Lsa	47	6	60
Config 11	0,4	Luhn	48	5	60
Config 12	0,4	TextRank	47	6	60
Config 13	0,2	LexRank	36	17	80
Config 14	0,2	Lsa	37	16	80
Config 15	0,2	Luhn	45	8	80
Config 16	0,2	TextRank	44	9	80

Tabela 5.3: Configurações de sumarização ao longo de todos os documentos considerados.

Analisando a Tabela 5.3 verificamos que para um rácio de 0.8, isto é, 80% do tamanho dos documentos originais, não existe diferença entre os algoritmos considerados, tendo todos detetado 49 VPs e apenas 4 FNs. Para o rácio de 0.6 os resultados são idênticos ao rácio de 0.8 concluindo-se não existir vantagens em sumarizar a 80% já que os resultados são iguais com uma sumarização de 60% sendo que o documento final é menor. A mesma lógica aplica-se para o rácio de 0.4 onde em três dos quatro algoritmos utilizados obtêm também eles resultados idênticos aos anteriores. Para um rácio de 0.2 destacam-se os algoritmos Luhn e TextRank que, com um ganho de 80%, conseguem detetar 45 e 44 VPs, respetivamente, num total de 53.

5.2.4 Sumarização + *SentenceFlags*

Por fim, a sumarização + *SentenceFlags* compreende uma abordagem que faz a extração de frases relevantes em cima de um documento sumariado 5.2.3. Para o efeito foi utilizada a configuração com os melhores resultados *SentenceFlags(has_ annotations=True, has_ keywords=None)*. A tabela 5.4 apresenta os resultados obtidos através desta abordagem.

	Racio	Algoritmo	Parâmetros Avaliação		
			VPs	FNs	Ganho (%)
Config 1	0,8	LexRank	40	13	46
Config 2	0,8	Lsa	40	13	46
Config 3	0,8	Luhn	40	13	46
Config 4	0,8	TextRank	40	13	46
Config 5	0,6	LexRank	40	13	58
Config 6	0,6	Lsa	40	13	58
Config 7	0,6	Luhn	40	13	58
Config 8	0,6	TextRank	39	14	57
Config 9	0,4	LexRank	28	25	70
Config 10	0,4	Lsa	38	15	70
Config 11	0,4	Luhn	39	14	70
Config 12	0,4	TextRank	39	14	69
Config 13	0,2	LexRank	24	29	84
Config 14	0,2	Lsa	28	25	85
Config 15	0,2	Luhn	37	16	84
Config 16	0,2	TextRank	37	16	84

Tabela 5.4: Configurações de sumarização + *SentenceFlags(has_annotations=True, has_keywords=None)* ao longo de todos os documentos considerados.

Comparando com a tabela da abordagem base, Tabela 5.3, verificamos que existe uma diminuição das frases relevadas detetadas (VP's) e um aumento do ganho. Considera-se no entanto este *trade off* negativo já que o aumento no ganho não justifica a diminuição dos VP's, principalmente para os resultados relativos a um rácio de 0,2.

5.2.5 Sumário

Ao longo do capítulo foram consideradas várias abordagens diferentes nomeadamente a utilização de *Sentence Flags* dadas pela combinação dos atributos *has_annotations* e *has_keywords*, similaridade semântica, sumarização e ainda e ainda sumarização + *SentenceFlags*. No geral, atendo à dificuldade da tarefa, considera-se que a maioria das abordagens consegue alcançar resultados razoáveis mas não necessariamente aceitáveis quando consideramos o objetivo do trabalho, fugindo à regra a sumarização. Tanto na *SentenceFlags* como na similaridade semântica existe um *trade off* nem sempre óbvio entre a ganho e o número de fases relevantes. Já na sumarização essa escolha torna-se mais óbvia obtendo-se resultados superiores tanto no ganho como na identificação de frases relevantes em comparação com as abordagens anteriores. A sumarização + *SentenceFlags* apesar de um incremento no ganho considera-se não justificar a diminuição das frases detetadas, sendo a sumarização só por si a melhor das abordagens testadas.

Capítulo 6

Conclusão

Através da extração automática de frases relevantes de documentos, neste trabalho foi desenvolvida uma ferramenta de apoio ao avaliador de segurança onde, pela utilização das frase extraídas, o avaliador deixa de necessitar de ler os documentos da íntegra, reduzindo-se o tempo de elaboração de cada relatório toxicológico.

Recorrendo a abordagens de sumarização a dimensão dos documentos é reduzida a 20% do seu tamanho original, sendo identificadas como relevantes 45 de 53 frases utilizadas numa abordagem manual, realizada por um especialista. Apesar de animadores os resultados têm de ser interpretados com reserva já que a avaliação é feita apenas de forma automática, levantando-se questões como: Não sendo todas as frases identificadas pelo algoritmo será necessário o avaliador ler o documento na íntegra?; Existe redundância nas anotações utilizadas? Se os documentos fossem anotados por outro avaliador as anotações seriam as mesmas?; Havendo informação em falta será que com a leitura de mais um documento se consegue ir buscar essa informação? Com o objetivo de aumentar a fiabilidade do programa são deixadas algumas abordagens novas como trabalho futuro tais como a utilização de relações entre entidades como atributo na tentativa de as associar a frases relevantes, a utilização de um modelo binário de classificação diferenciando as frases relevantes das frases não relevantes e ainda, não sendo uma abordagem mas indo de encontro ao referido acima, uma avaliação manual dos resultados de maneira a obter-se uma validação efetiva, real, da viabilidade das abordagens em estudo.

Referências

[1] Saber Akhondi, Alexander Klenner, Christian Tyrchan, Anil Manchala, Kiran Boppana, Daniel Lowe, Marc Jacobs, J. Sarma, Roger Sayle, Jan Kors, and Sorel Muresan. Annotated chemical patent corpus: A gold standard for text mining. PloS one, 9:e107477, 09 2014.

[2] Waleed Ammar, Dirk Groeneveld, Chandra Bhagavatula, Iz Beltagy, Miles Crawford, Doug Downey, Jason Dunkelberger, Ahmed Elgohary, Sergey Feldman, Vu Ha, Rodney Kinney, Sebastian Kohlmeier, Kyle Lo, Tyler Murray, Hsu-Han Ooi, Matthew Peters, Joanna Power, Sam Skjonsberg, Lucy Wang, Chris Wilhelm, Zheng Yuan, Madeleine van Zuylen, and Oren Etzioni. Construction of the literature graph in semantic scholar. In Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 3 (Industry Papers), pages 84–91, New Orleans - Louisiana, June 2018. Association for Computational Linguistics.

[3] Michael Bada, Miriam Eckert, Donald Evans, Kristin Garcia, Krista Shipley, Dmitry Sitnikov, William Baumgartner Jr, Kevin Cohen, Karin Verspoor, Judith Blake, and Lawrence Hunter. Concept annotation in the craft corpus. BMC bioinformatics, 13:161, 07 2012.

[4] Iz Beltagy, Kyle Lo, and Arman Cohan. Scibert: A pretrained language model for scientific text, 2019.

[5] Nigel Collier and Jin-Dong Kim. Introduction to the bio-entity recognition task at JNLPBA. In Proceedings of the International Joint Workshop on Natural Language Processing in Biomedicine and its Applications (NLPBA/BioNLP), pages 73–78, Geneva, Switzerland, August 28th and 29th 2004. COLING.

[6] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding, 2019.

[7] J Ding, Daniel Berleant, D Nettleton, and Eve Wurtele. Mining medline: abstracts, sentences, or phrases? Pacific Symposium on Biocomputing. Pacific Symposium on Biocomputing, 7:326–37, 02 2002.

[8] Rezarta Dogan, Robert Leaman, and Zhiyong lu. Ncbi disease corpus: A resource for disease name recognition and concept normalization. Journal of biomedical informatics, 47, 01 2014.

[9] Rezarta Dogan, Robert Leaman, and Zhiyong lu. Ncbi disease corpus: A resource for disease name recognition and concept normalization. Journal of biomedical informatics, 47, 01 2014.

[10] Martin Gerner, Goran Nenadic, and Casey Bergman. Linnaeus: A species name identification system for biomedical literature. BMC bioinformatics, 11:85, 02 2010.

[11] John M Giorgi and Gary D Bader. Transfer learning for biomedical named entity recognition with neural networks. Bioinformatics, 34(23):4087–4094, 06 2018.

[12] Tatyana Goldberg, Shrikant Vinchurkar, Juan Miguel Cejuela, Lars Jensen, and Burkhard Rost. Linked annotations: a middle ground for manual curation of biomedical databases and text corpora. BMC Proceedings, 9:A4, 08 2015.

[13] Udo Hahn, Katrin Tomanek, Elena Beisswanger, and Erik Faessler. A proposal for a configurable silver standard. ACL 2010 - LAW 2010: 4th Linguistic Annotation Workshop, Proceedings, pages 235–242, 01 2010.

[14] Marti A. Hearst. Automatic acquisition of hyponyms from large text corpora. In COLING 1992 Volume 2: The 14th International Conference on Computational Linguistics, 1992.

[15] Şenay Kafkas, Ian Lewin, David Milward, Erik van Mulligen, Jan Kors, Udo Hahn, and Dietrich Rebholz-Schuhmann. CALBC: Releasing the final corpora. In Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12), pages 2923–2926, Istanbul, Turkey, May 2012. European Language Resources Association (ELRA).

[16] Shweta Bagewadi Kawalia, Tamara Bobić, Martin Hofmann-Apitius, Juliane Fluck, and Roman Klinger. Detecting mirna mentions and relations in biomedical literature. F1000Research, 03 2015.

[17] Martin Krallinger, Obdulia Rabal, Florian Leitner, Miguel Vazquez, David Salgado, Zhiyong lu, Robert Leaman, Yanan Lu, Donghong Ji, Daniel Lowe, Roger Sayle, Riza Batista-Navarro, Rafal Rak, Torsten Huber, Tim Rocktäschel, Sérgio Matos, David Campos, Buzhou Tang, Wang Qi, and Alfonso Valencia. The chemdner corpus of chemicals and drugs and its annotation principles. Journal of Cheminformatics, 7:S2, 03 2015.

[18] Taku Kudo and John Richardson. Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing, 2018.

[19] Robert Leaman, C. Miller, and G. Gonzalez. Enabling recognition of diseases in biomedical text with machine learning: corpus and benchmark. Proceedings of the 2009 Symposium on Languages in Biology and Medicine, pages 82–89, 01 2009.

[20] Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. Biobert: a pre-trained biomedical language representation model for biomedical text mining. Bioinformatics, Sep 2019.

[21] Jiao Li, Yueping Sun, Robin Johnson, Daniela Sciaky, Chih-Hsuan Wei, Robert Leaman, Allan Peter Davis, Carolyn Mattingly, Thomas Wiegers, and Zhiyong lu. Biocreative v cdr task corpus: a resource for chemical disease relation extraction. Database, 2016:baw068, 05 2016.

[22] Jiao Li, Yueping Sun, Robin Johnson, Daniela Sciaky, Chih-Hsuan Wei, Robert Leaman, Allan Peter Davis, Carolyn Mattingly, Thomas Wiegers, and Zhiyong lu. Biocreative v cdr task corpus: a resource for chemical disease relation extraction. Database, 2016:baw068, 05 2016.

[23] Mike Mintz, Steven Bills, Rion Snow, and Daniel Jurafsky. Distant supervision for relation extraction without labeled data. In Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on

Natural Language Processing of the AFNLP, pages 1003–1011, Suntec, Singapore, August 2009. Association for Computational Linguistics.

[24] Mariana Neves, Alexander Damaschun, Andreas Kurtz, and Ulf Leser. Annotating and evaluating text for stem cell research. 01 2012.

[25] Philip Ogren, Guergana Savova, and Christopher Chute. Constructing evaluation corpora for automated clinical named entity recognition. In Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC'08), Marrakech, Morocco, May 2008. European Language Resources Association (ELRA).

[26] Evangelos Pafilis, Sune Frankild, Lucia Fanini, Sarah Faulwetter, Christina Pavloudi, Katerina Vasileiadou, C. Arvanitidis, and Lars Jensen. The species and organisms resources for fast and accurate identification of taxonomic names in text. PloS one, 8:e65390, 06 2013.

[27] Gorjan Popovski, Barbara Seljak, and Tome Eftimov. A survey of named-entity recognition methods for food information extraction. IEEE Access, 8:31586–31594, 02 2020.

[28] Sampo Pyysalo, Filip Ginter, Juho Heimonen, Jari Björne, Jorma Boberg, Jouni Järvinen, and Tapio Salakoski. Bioinfer: A corpus for information extraction in the biomedical domain. BMC bioinformatics, 8:50, 02 2007.

[29] Sampo Pyysalo, Tomoko Ohta, and Sophia Ananiadou. Overview of the cancer genetics (CG) task of BioNLP shared task 2013. In Proceedings of the BioNLP Shared Task 2013 Workshop, pages 58–66, Sofia, Bulgaria, August 2013. Association for Computational Linguistics.

[30] Lance Ramshaw and Mitch Marcus. Text chunking using transformation-based learning. In Third Workshop on Very Large Corpora, 1995.

[31] Guergana Savova, James Masanz, Philip Ogren, Jiaping Zheng, Sunghwan Sohn, Karin Kipper-Schuler, and Christopher Chute. Mayo clinical text analysis and knowledge extraction system (ctakes): Architecture, component evaluation and applications. Journal of the American Medical Informatics Association : JAMIA, 17:507–13, 09 2010.

[32] Mike Schuster and Kaisuke Nakajima. Japanese and korean voice search, 2012.

[33] Rico Sennrich, Barry Haddow, and Alexandra Birch. Neural machine translation of rare words with subword units, 2016.

[34] L. Smith, L. Tanabe, R. Ando, C. Kuo, I-Fang Chung, C. Hsu, Y. Lin, R. Klinger, Christoph Friedrich, K. Ganchev, M. Torii, Hongfang Liu, Barry Haddow, Craig Struble, Richard Povinelli, Andreas Vlachos, William Baumgartner Jr, Lawrence Hunter, B. Carpenter, and W. Wilbur. Overview of biocreative ii gene mention recognition. Genome Biology, 9, 09 2008.

[35] Karin Verspoor, Antonio Jimeno-Yepes, Lawrence Cavedon, Tara McIntosh, Asha Herten-Crabb, Zoë Thomas, and John-Paul Plazzer. Annotating the biomedical literature for the human variome. Database : the journal of biological databases and curation, 2013:bat019, 01 2013.

[36] Xinglong Wang, Jun'ichi Tsujii, and Sophia Ananiadou. Disambiguating the species of biomedical named entities using natural language parsers. Bioinformatics (Oxford, England), 26:661–7, 03 2010.

[37] Leon Weber, Jannes Münchmeyer, Tim Rocktäschel, Maryam Habibi, and Ulf Leser. HUNER: improving biomedical NER with pretraining. Bioinformatics, 36(1):295–302, 06 2019.

[38] Leon Weber, Mario Sänger, Jannes Münchmeyer, Maryam Habibi, Ulf Leser, and Alan Akbik. Hunflair: An easy-to-use tool for state-of-the-art biomedical named entity recognition. CoRR, abs/2008.07347, 2020.

[39] Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V. Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, Jeff Klingner, Apurva Shah, Melvin Johnson, Xiaobing Liu, Łukasz Kaiser, Stephan Gouws, Yoshikiyo Kato, Taku Kudo, Hideto Kazawa, Keith Stevens, George Kurian, Nishant Patil, Wei Wang, Cliff Young, Jason Smith, Jason Riesa, Alex Rudnick, Oriol Vinyals, Greg Corrado, Macduff Hughes, and Jeffrey Dean. Google's neural machine translation system: Bridging the gap between human and machine translation, 2016.

[40] Yukun Zhu, Ryan Kiros, Richard Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books, 2015.

Appendices

Apêndice A: *Gold Standard Corporas*

Entity Type	Corpus	Text genre	Text type	No. sentences	No. tokens	No. unique tokens	No. annotations	No. unique annotations
Chemicals	BioSemantics [1]	Patent	Full-text	163219	6608020	173193	386110	72782
	CDR [22]	Scientific Article	Abstract	14166	326506	22083	15915	2623
	CHEMDNER patent [17]	Patent	Abstract	35679	1495524	60850	65685	20630
Diseases	Arizona Disease [19]	Scientific	Article	2804	74346	8133	3425	1266
	CDR [22]	Scientific	Article	14166	326506	22083	12617	3113
	miRNA [16]	Scientific	Article	2676	66419	7638	2159	606
	NCBI Disease [9]	Scientific	Article	7645	173283	12534	6881	2136
	Variome [35]	Scientific	Article	6274	177119	12307	5904	451
Species	CellFinder [24]	Scientific	Article	2234	66519	7584	479	42
	Linneaus [10]	Scientific	Article	19048	491253	33132	4259	419
	LocText [12]	Scientific	Article	956	22756	4335	276	37
	miRNA [16]	Scientific	Article	2676	66419	7638	722	41
	S800 [26]	Scientific	Article	8356	198091	19992	3708	1503
	Variome [35]	Scientific	Article	6274	177119	12307	182	8
Genes/proteins	BioCreative II GM [34]	Scientific	Article	20384	514146	49365	24596	15841
	BioInfer [28]	Scientific	Article	1147	34187	5200	4378	1041
	CellFinder [24]	Scientific	Article	2234	66519	7584	1750	734
	DECA [36]	Scientific	Article	5492	139771	14053	6324	2127
	FSU-PRGE [13]	Scientific	Article	35361	914717	453634	59489	27363
	IEPA [7]	Scientific	Article	241	15365	2871	1110	130
	LocText [12]	Scientific	Article	956	22756	4335	1395	549
	miRNA [16]	Scientific	Article	2676	66419	7638	1058	370
	Variome [35]	Scientific	Article	6274	177119	12307	4617	458

Tabela 1: Gold standard corpora (retirado de [11])