**University of Minho**
School of Engineering

# Trabalho Prático 2
# Plataforma para Visualização de Conceitos Médicos

Processamento de Linguagem Natural

Daniel Sá (PG56120)

João Cardoso (PG56135)

Jorge Costa (PG56136)

Pedro Flores (A100475)

june 2025

# Índice

# 1 Contextualização

No âmbito da unidade curricular de Processamento de Linguagem Natural, inserida no curso de Engenharia Biomédica, foi desenvolvido um sistema que dá continuidade ao trabalho prático anterior, onde foram extraídos dados em formato JSON a partir de documentos médicos em PDF. Este projeto teve como principal objetivo enriquecer a informação anteriormente adquirida, recorrendo a fontes externas como websites e dicionários médicos especializados. Através deste enriquecimento, foram adicionados novos conceitos e ampliados os já existentes com atributos relevantes como sinónimos, descrições, categorias e relações semânticas.

Para permitir a exploração eficaz deste novo conjunto de dados, foi implementada uma plataforma interativa que possibilita a visualização e manipulação do dataset de forma intuitiva. Esta ferramenta permite não só a navegação entre conceitos e a análise das suas inter-relações, como também oferece funcionalidades de adição, atualização e remoção de termos de forma persistente, assegurando assim a manutenção contínua da base de dados.

A motivação para este trabalho reside na crescente importância da organização e processamento automático da informação médica, contribuindo para melhorar o acesso ao conhecimento clínico e apoiar a tomada de decisões na área da saúde. Assim, este projeto procura facilitar a utilização e interpretação de dados médicos complexos, através de uma abordagem robusta, flexível e centrada na usabilidade.

Este relatório tem como finalidade documentar as tecnologias utilizadas, bem como a arquitetura do sistema implementado, detalhando o processo de desenvolvimento e as soluções adotadas.

# 2 Web Scraping

Com o intuito de integrar novos conceitos e enriquecer os previamente extraídos, nomeadamente os relacionados com doenças, foi implementado um sistema automatizado de recolha de informação a partir de fontes externas. Em particular, recorreu-se à técnica de web scraping sobre o portal Atlas da Saúde, uma fonte fidedigna de informação médica em português.

Através de um script desenvolvido em Python, utilizando as bibliotecas requests e BeautifulSoup, foi realizada a navegação automática pelas diferentes páginas da secção "Doenças de A a Z" do website. Para cada letra do alfabeto, foram extraídas as designações das doenças e os respetivos resumos informativos. Estes dados foram armazenados num dicionário estruturado, em que cada chave corresponde ao nome da doença e o valor associado contém a respetiva descrição.

Posteriormente, todo o conteúdo recolhido foi consolidado num único ficheiro JSON, garantindo a persistência e organização da informação de forma estruturada. Este ficheiro foi então integrado no sistema desenvolvido, permitindo complementar os conceitos do dataset original com descrições detalhadas diretamente obtidas de uma fonte médica confiável.

Este processo de scraping automatizado permitiu não só aumentar significativamente a quantidade de informação disponível, mas também melhorar a qualidade e profundidade do conhecimento associado a cada termo, contribuindo para uma visualização mais rica e contextualizada dos dados.

# 3 Agregação da Informação

As principais nuances do processo de agregação de dados foram já abordadas no primeiro trabalho prático. No entanto, tendo em conta a forte dependência dos dados neste projeto e a introdução de novos conceitos provenientes do processo de web scraping, tornou-se pertinente revisitar e refinar a estratégia de agregação inicialmente desenvolvida. Esta revisão visou, sobretudo, melhorar a qualidade dos dados no contexto da língua portuguesa e garantir uma estrutura mais consistente e adequada para utilização numa aplicação web.

A nova estratégia de agregação tem início mesmo antes da concatenação dos ficheiros JSON criados na primeira fase do projeto, com uma reestruturação do dicionário multilingue da COVID-19. Este dicionário foi adaptado para priorizar entradas em português — algo que, na versão anterior, apenas era considerado na fase final de fusão dos dados, abrindo margem para erros e perdas de informação potencialmente relevantes. Nesta nova abordagem, apenas foram mantidas as entradas com termos em português, e parâmetros como "sin."e "sin. compl."foram consolidados num novo campo denominado "sinonimos_ca", distinguindo explicitamente os sinónimos em catalão, a língua original do dicionário, cuja relevância foi diminuída na estrutura final.

Adicionalmente, o JSON relativo ao dicionário da COVID-19 foi sujeito a um pequeno ajuste que havia sido negligenciado no trabalho anterior: a separação do tipo de descrição em catalão da própria descrição, clarificando o conteúdo e a sua origem linguística.

Com esta reformulação, a agregação dos restantes ficheiros JSON passou a ser realizada tendo este dicionário como base, uma vez que apresenta maior densidade e riqueza de informação por entrada, comparativamente aos restantes. A fusão dos dados seguiu os mesmos princípios adotados anteriormente: priorização de entradas em português, preservação de todas as descrições portuguesas existentes para cada termo e inclusão de sinónimos, traduções e siglas ausentes no dicionário de base. Foi também nesta fase que se introduziram novos conceitos identificados através do processo de web scraping.

O script desenvolvido para esta tarefa automatiza todo este processo, incluindo a fusão controlada de dicionários com base em correspondência de termos (insensível a maiúsculas/minúsculas), gestão de duplicados em traduções, preservação de múltiplas descrições e unificação de estruturas divergentes. Esta abordagem permitiu gerar um dicionário final mais robusto, expressivo e adequado às necessidades exploratórias da aplicação desenvolvida.

# 4 Categorização de Conceitos

Para organizar sistematicamente o extenso léxico de termos médicos, foi implementada uma metodologia semi-automatizada, em duas fases. O objetivo principal era agrupar milhares de termos individuais num conjunto finito de categorias coerentes e significativas. Esta abordagem foi concebida para aproveitar o poder da aprendizagem automática para um processamento de dados objetivo, incorporando simultaneamente a revisão humana para o refinamento e validação finais.

O processo divide-se em duas fases principais:

- **Agrupamento semântico automatizado:** Utilização de um modelo linguístico para compreender o significado dos termos e agrupá-los com base na semelhança.
- **Revisão e refinamento manual:** Identificação manual dos clusters gerados e consolidação dos mesmos numa estrutura final e lógica.

Para agrupar os termos médicos, esta fase inicial baseou-se no processamento de linguagem natural (PNL) e em algoritmos de agrupamento. A pedra basilar da abordagem foi a conversão de cada termo e da sua descrição numa representação numérica, conhecida como embedding. Isto foi conseguido utilizando o modelo `neuralmind/bert-base-portuguese-cased`, uma variante do BERT pré-treinada em texto português. Ao contrário da simples correspondência de palavras-chave, estes embeddings captam o significado semântico, permitindo ao sistema reconhecer que termos como "enfarte do miocárdio" e "ataque cardíaco" são conceptualmente semelhantes porque as suas representações vetoriais são próximas no espaço multidimensional.

Antes de o agrupamento efetivo ocorrer, foi necessário determinar o número ideal de categorias. Para isso, foi utilizado o método do 'Elbow'. O algoritmo de agrupamento K-Means foi executado iterativamente com um número crescente de clusters e, para cada execução, a soma de quadrados dentro do cluster (WCSS) foi calculada para medir a compactação do cluster. Ao traçar o WCSS em relação ao número de clusters, forma-se tipicamente um ponto de "cotovelo" distinto. Este ponto significa o compromisso ideal, em que a adição de mais clusters proporciona retornos decrescentes na compactidade, oferecendo assim uma forma orientada por dados para selecionar o número adequado de categorias iniciais.

Com o número ótimo de clusters identificado, o algoritmo K-Means foi aplicado ao conjunto de todos os termos incorporados. O K-Means funciona dividindo estes vectores no número especificado de clusters, atribuindo cada termo ao cluster com o "centróide" mais próximo, que é a média geométrica de todos os termos desse grupo. Isto resultou naturalmente na formação de grupos semanticamente relacionados,

separando efetivamente os termos relacionados com anatomia, doenças e medicamentos em grupos distintos

Após o agrupamento automático, o processo entrou numa segunda fase centrada na revisão e no melhoramento. O resultado inicial consistia em grupos numerados que, embora internamente coerentes, necessitavam de interpretação e validação humanas para serem úteis.

O primeiro passo foi analisar os termos mais representativos dentro de cada grupo gerado, os mais próximos do centróide do grupo, para compreender o seu tema central. Com base nesta análise, foi atribuído manualmente um nome descritivo e legível por humanos a cada grupo. Por exemplo, um grupo que continha termos como "ensovibep", "zilucoplan" e 'acalabrutinib' foi apropriadamente rotulado de "Fármacos e Terapêuticas".

Durante esta revisão, também se tornou evidente que o algoritmo tinha criado alguns grupos semanticamente semelhantes. Por exemplo, surgiram grupos distintos para "Farmacologia", "Classes de Fármacos" e "Fármacos e Terapêuticas". Para estabelecer uma hierarquia mais lógica e menos granular, estes grupos estreitamente relacionados foram consolidados manualmente numa única categoria mais ampla, como "Farmacologia e Terapêutica". Este refinamento garantiu que a categorização final fosse significativa e intuitivamente estruturada.

O resultado final deste processo de duas fases é um ficheiro JSON que contém um dicionário limpo e estruturado. Neste ficheiro, cada chave é um nome de categoria final, legível por humanos, e o seu valor é a lista completa de termos médicos que lhe pertencem. Esta metodologia semi-automatizada combina a objetividade e a escalabilidade da aprendizagem automática com o conhecimento de domínio crucial de um ser humano, garantindo uma categorização robusta do léxico médico.

# 5 Rotas do Backend

Esta secção do relatório tem como finalidade detalhar a arquitetura e a implementação das rotas do backend do sistema desenvolvido, focando na sua construção utilizando o framework Flask para Python. As rotas são os pontos de entrada da aplicação web, responsáveis por processar as requisições dos utilizadores, interagir com o dataset de dados médicos enriquecido e retornar as respostas adequadas, muitas vezes sob a forma de páginas HTML renderizadas.

Serão exploradas as funcionalidades de cada rota, desde a apresentação da página inicial até às operações CRUD (Criar, Ler, Atualizar, Eliminar) sobre os conceitos médicos, bem como mecanismos avançados como a hiperligação dinâmica entre conceitos e a apresentação de estatísticas. A descrição de cada rota incluirá o seu propósito, os parâmetros de entrada, as lógicas de processamento e a sua contribuição para a usabilidade e manutenção da base de conhecimento. A persistência das alterações realizadas através das rotas será igualmente abordada, sublinhando a importância da integridade e da atualidade dos dados.

- **Página inicial:** Esta rota, definida como a raiz da aplicação (`/`), tem como principal objetivo a apresentação da página inicial do projeto. Ao ser acedida, a função `home()` é invocada e retorna o template `home.html`. Embora o seu propósito seja aparentemente simples, a página inicial é crucial para fornecer uma porta de entrada intuitiva ao utilizador, servindo como o ponto de partida para a exploração das funcionalidades do sistema. Esta rota estabelece o contexto visual e de navegação para as demais funcionalidades da plataforma web.

- **Tabela:** A rota (`/tabela`) para a funcionalidade da tabela foi concebida para lidar eficientemente com o processamento e a filtragem de dados. O processo começa com o carregamento do dicionário principal `merged_dict.json` e os termos categorizados `categorized_medical_terms.json` na memória quando a aplicação é iniciada. Essa estratégia de pré-carregamento garante que os dados estejam prontamente acessíveis sem a necessidade de ler os arquivos em cada solicitação, o que melhora significativamente o desempenho.

  O núcleo da lógica de filtragem reside no script `filters.py`, que contém um conjunto de funções, cada uma concebida para filtrar o dicionário com base num critério específico. Estas funções cobrem uma vasta gama de condições, tais como a presença ou ausência de descrições, a disponibilidade de traduções para uma língua específica, a letra inicial do termo e o facto de um termo ser simples ou composto. Os parâmetros de consulta do pedido Web de um utilizador são interpretados para decidir quais os termos que devem ser apresentados. Além disso, antes de

serem enviados para o frontend, os resultados filtrados são verificados para ver se uma consulta de pesquisa está presente; se estiver, é aplicada uma função de realce ao termo e à sua descrição, envolvendo o texto correspondente em etiquetas `<strong>` para o fazer sobressair para o utilizador.

- **Detalhes:** A rota de detalhes (`/detalhes/<string:conceito>`) tem como principal função apresentar toda a informação disponível sobre um determinado termo. Para além da simples exibição de atributos como a descrições, traduções, sinónimos, categorias, siglas, etc, esta rota implementa um mecanismo adicional que enriquece a experiência de navegação: um sistema de hiperligação dinâmica entre conceitos.

  Através de uma função auxiliar, o conteúdo textual (nomeadamente descrições em português, abonações e excertos de enciclopédias) é analisado à procura de outras palavras-chave que coincidam com conceitos já existentes no dataset. Sempre que uma correspondência é encontrada, essa palavra é convertida num link clicável. Ao carregar nessa palavra, o utilizador é automaticamente redirecionado para a página de detalhes do termo correspondente, permitindo uma navegação contextual e fluida entre conceitos relacionados.

- **Adicionar:** Para permitir ao utilizador expandir o conhecimento representado no sistema, foi criada uma rota dedicada à adição de novos conceitos (`/adicionar`). Esta rota é responsável por receber e processar os dados submetidos através de um formulário na interface frontend. Os dados incluem atributos como o nome do termo, descrições, sinónimos, siglas, etc .

  Após validação e tratamento da informação recebida, os dados são inseridos de forma estruturada no dataset principal, garantindo a sua persistência mesmo após reinicializações do sistema. Não é permitida a adição de termos já existentes no datsaset. Esta funcionalidade permite que a base de conhecimento evolua continuamente, incorporando novos elementos de forma simples e intuitiva.

- **Editar:** A funcionalidade de edição permite ao utilizador atualizar ou corrigir a informação associada a um conceito já existente (`/editar/<string:conceito>`). Esta rota é ativada a partir de um formulário presente na interface gráfica, onde o utilizador pode modificar atributos como descrições, sinónimos, categorias lexicais, traduções, etc.

  O backend processa os dados recebidos, verifica a existência do termo no dataset e substitui os atributos antigos pelos novos. Esta atualização é feita de forma permanente, assegurando que qualquer modificação se reflete imediatamente na visualização e fica guardada no armazenamento

persistente. A funcionalidade de edição é essencial para manter a coerência, precisão e atualidade da base de dados.

- **Remover:** De modo a eliminar a informação desnecessária ou incorreta do dataset, esta função pode ser acedida na rota (`/detalhes/<string:conceito>`) partir da interface gráfica. Ao selecionar esta função, o utilizador receberá um pop-up para confirmar a ação.

  Recebendo a confirmação, o conceito é removido do dataset de forma permanente, garantindo que tais modificações sejam guardadas no armazenamento persistente. Tal como as outras funções de manipulação de dados, a possibilidade de remover conceitos é essencial para a manutenção e organização do dataset.

- **Estatísticas:** O projeto inclui uma funcionalidade dedicada à geração e disponibilização de estatísticas gerais sobre o dicionário construído. Esta rota tem como objetivo principal fornecer dados sumarizados para utilização em templates HTML, permitindo uma visualização rápida e informativa da composição e cobertura do dicionário. As estatísticas são calculadas por um script no backend, que processa o ficheiro JSON consolidado e deriva automaticamente um conjunto de métricas descritivas.

  O script calcula o número total de entradas e decompõe este total para mostrar quantas entradas incluem descrições, traduções e sinónimos. Além disso, distingue entre termos simples e compostos, oferecendo tanto contagens brutas como percentagens para cada categoria. Isto permite uma avaliação rápida da exaustividade do dicionário e da riqueza dos dados disponíveis para cada termo.

# 6 Interface do Utilizador

Para proporcionar uma interface visual intuitiva e interativa aos utilizadores, o frontend do sistema foi desenvolvido utilizando um conjunto de páginas HTML dinâmicas, potenciadas pelo motor de templates Jinja2. Esta abordagem permite a criação de conteúdo web flexível e reutilizável. Todas as páginas da aplicação partilham uma estrutura comum, tendo como base o template `layout.html`. Este template principal define a estrutura geral do website, incluindo elementos como a barra de navegação e a inclusão stylesheets e scripts JavaScript, garantindo assim uma experiência de utilizador consistente e uma manutenção eficiente do código em toda a aplicação.

- **Página inicial:** A página inicial do projeto serve como o ponto de entrada principal para os utilizadores. Visualmente, é composta pelo template `home.html`, garantindo a presença de uma barra de navegação no topo para facilitar a navegação entre as diferentes funcionalidades do sistema. O conteúdo central da página exibe uma imagem que representa a instituição escolar, acompanhada de uma mensagem de boas-vindas e uma breve descrição que contextualiza o projeto como parte da disciplina de Processamento de Linguagem Natural. Esta combinação de elementos visuais e textuais proporciona uma introdução clara e intuitiva ao utilizador.

- **Tabela:** O template `table.html` oferece uma interface altamente interativa e de fácil utilização para a exploração dos dados do dicionário. A página organiza-se em torno de uma tabela central que apresenta os conceitos, as respetivas descrições e um botão "Detalhes" que permite aceder à entrada completa. A principal interação do utilizador ocorre através de um conjunto de controlos de filtragem posicionados diretamente acima da tabela. Entre as opções disponíveis, destaca-se a pesquisa de texto com suporte a expressões regulares e sensibilidade a maiúsculas, permitindo buscas avançadas e mais precisas. Adicionalmente, é possível filtrar os termos por categoria médica ou por disponibilidade de tradução numa determinada língua. Estes filtros visíveis oferecem um acesso rápido e direto às necessidades de pesquisa mais comuns.

  Para um controlo mais granular, a interface inclui uma janela modal para "Filtros avançados". Este modal contém opções de filtragem mais específicas, tais como mostrar apenas termos com (ou sem) descrições, filtrar pela primeira letra do termo, especificar um comprimento mínimo ou máximo de caracteres e distinguir entre termos simples e compostos. As seleções deste modal são submetidas juntamente com os filtros principais, permitindo consultas complexas e em camadas.

- **Detalhes:** Esta página do frontend tem como objetivo apresentar, de forma estruturada e acessível, todas as características associadas a um conceito específico enviado pelo backend. A interface

exibe campos como definições, sinónimos, categorias e descrições, etc, organizados de forma clara para facilitar a consulta. Apenas os campos que possuem informação são demonstrados.

Além da visualização passiva da informação, a página disponibiliza botões que permitem ao utilizador editar ou remover o termo do dataset. Estas ações estão integradas com o backend através de chamadas dinâmicas e respeitam a persistência dos dados.

- **Adicionar:** Através desta página, o utilizador pode introduzir novos conceitos na base de dados do sistema. A página apresenta um formulário dinâmico e reativo que permite inserir diversos atributos do conceito, como nome, descrições, categorias, sinónimos, etc.

  Os campos do formulário são organizados com base em grupos lógicos, e apenas alguns deles são obrigatórios, permitindo flexibilidade na introdução dos dados. O código HTML implementa verificações condicionais que associam certos campos entre si (por exemplo, certos campos opcionais exigem a presença de outros para garantir coerência).

  Após a submissão do formulário, os dados são enviados para a rota correspondente do backend, onde são processados e armazenados. Concluído este processo, o utilizador é automaticamente redirecionado para a página de detalhes do novo conceito, promovendo uma experiência contínua e integrada.

- **Editar:** Semelhante em estrutura à página de adição, a página de edição distingue-se por apresentar o formulário já pré-preenchido com os dados atuais do termo selecionado. Desta forma, o utilizador pode tanto modificar informações existentes como adicionar novos dados relacionados ao conceito.

  A reutilização do mesmo modelo de formulário facilita a manutenção do código e a experiência do utilizador. Após a edição, os dados são enviados ao backend, atualizando o registo correspondente de forma persistente. Tal como na adição, o utilizador é redirecionado para a página de detalhes, onde poderá confirmar as alterações efetuadas de imediato.

- **Estatísticas:** A página de estatísticas apresenta uma visão geral clara e interativa sobre o conteúdo do dicionário. Alimentada pelos dados fornecidos pela rota dedicada no backend, esta página estrutura visualmente a informação para facilitar a interpretação por parte do utilizador. O seu principal objetivo é permitir uma análise intuitiva da qualidade dos dados existentes e apoiar a identificação de áreas onde a base lexical pode ser enriquecida.

  A visualização foi concebida no modelo `stats.html` com foco na simplicidade e impacto. Quatro

métricas principais — número total de entradas, entradas com descrições, com traduções e com sinónimos — são destacadas em cartões informativos bem definidos. Esta visão de alto nível é complementada por barras percentuais que ilustram a proporção de termos com e sem cada um destes atributos, oferecendo uma leitura rápida da cobertura dos dados.

Para além disso, a interface exibe informação sobre a diversidade e estrutura do dicionário, incluindo a lista de línguas disponíveis para tradução, as categorias gramaticais identificadas e uma tabela que representa a distribuição dos termos pelas letras do alfabeto. Esta abordagem reforça a utilidade da página como ferramenta de diagnóstico e acompanhamento da evolução do conteúdo lexical.

# 7 Potenciais Melhorias

Tal como em qualquer projeto de desenvolvimento de software, é fundamental identificar oportunidades de melhoria que possam contribuir para a evolução contínua da aplicação. Embora a solução desenvolvida já integre um conjunto relevante de funcionalidades para a visualização dos dados recolhidos e um mecanismo robusto de categorização, existem áreas que, se aprimoradas, poderão melhorar significativamente tanto a qualidade dos dados utilizados como a capacidade exploratória da aplicação.

Com base no conjunto de dados utilizado na construção da aplicação, destaca-se o elevado potencial de enriquecimento da informação, uma vez que, após os processos de limpeza e associação entre diferentes dicionários, apenas cerca de 3000 entradas permaneceram. Um volume de dados mais expressivo aumentaria substancialmente as possibilidades de associação entre termos, permitindo a inclusão de um maior número de sinónimos, traduções e descrições alternativas. Isto contribuiria para uma base de conhecimento mais sólida, estruturada, fiável e completa.

Na prática, esta limitação foi parcialmente mitigada através da utilização de estratégias de web scraping, que permitiram recolher dados de diferentes fontes e aumentar o volume inicial de entradas. No entanto, estas estratégias poderiam ser expandidas e refinadas para recolher um conjunto ainda mais abrangente de dados, não só aumentando a diversidade de termos presentes no dicionário, como também enriquecendo semanticamente os conceitos existentes — por exemplo, através da introdução de mais sinónimos, traduções e descrições alternativas para os mesmos termos.

Atualmente, a aplicação recorre a um modelo BERT treinado com dados em português para a construção de word embeddings, que são posteriormente agrupados através do algoritmo de clustering K-means. No entanto, a nomeação dos clusters é realizada manualmente após a execução do algoritmo. Embora esta abordagem seja eficaz para o volume de dados atual, num cenário de escalabilidade — com um crescimento significativo do número de categorias — seria vantajoso explorar estratégias de nomeação automática baseadas em técnicas de aprendizagem automática, permitindo maior autonomia e consistência no processo.

Por fim, embora com impacto reduzido no contexto atual, o desempenho da aplicação poderia beneficiar de uma integração mais eficiente dos filtros de pesquisa diretamente no frontend, especialmente através de uma melhor utilização do plugin DataTables. Presentemente, o motor de busca implementado em Python oferece uma pesquisa robusta com suporte a múltiplos filtros, incluindo expressões regulares. No entanto, dada a arquitetura da aplicação, cada pesquisa requer um novo pedido (fetch) ao servidor, o que poderá ser evitado através de mecanismos de caching no lado do cliente — uma abordagem viável, tendo em conta que os dados não apresentam um volume excessivo.

# 8 Conclusão

A implementação deste programa de desenvolvimento do dataset permitiu uma expansão significativa da informação disponível, possibilitando não só a sua organização sistemática, como também uma exploração mais aprofundada dos conteúdos. Através da integração de múltiplas funcionalidades, o projeto consolidou-se como uma ferramenta robusta para a análise semântica dos termos e para a visualização estruturada dos dados.

Destacam-se, entre as funcionalidades desenvolvidas, a possibilidade de manipulação direta do conteúdo por parte do utilizador — com operações persistentes de adição, edição e remoção de conceitos — e a aplicação de técnicas de web scraping para a recolha automatizada de informação a partir de fontes externas fiáveis. Estas técnicas contribuíram para o enriquecimento do dataset, complementando os termos existentes com descrições detalhadas, sinónimos e classificações temáticas.

Além disso, a adoção de abordagens de agrupamento semântico, como a criação de clusters, permitiu identificar padrões e relações relevantes entre conceitos, promovendo uma organização mais coerente do conteúdo lexical. Neste processo, a gestão cuidada dos dados revelou-se essencial, assegurando a integridade, consistência e acessibilidade da informação. Um dataset bem estruturado não só facilita a navegação e visualização, como sustenta análises mais rigorosas, potenciando o uso de modelos como o BERT para a identificação automática de categorias e relações em contextos linguísticos complexos.