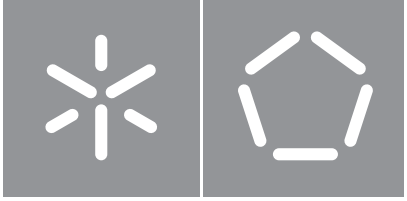
Universidade do Minho
Escola de Engenharia

Ricardo Filipe Sousa Caçador

Anotação Automática de Informação Clínica

Janeiro, 2022



Universidade do Minho
Escola de Engenharia

Ricardo Filipe Sousa Caçador

Anotação Automática de Informação Clínica

Dissertação de Mestrado
Mestrado Integrado em Engenharia Informática

Trabalho efetuado sob a orientação do
Professor Doutor Paulo Jorge Freitas de Oliveira Novais

Janeiro, 2022

DIREITOS DE AUTOR E CONDIÇÕES DE UTILIZAÇÃO DO TRABALHO POR TERCEIROS

Este é um trabalho académico que pode ser utilizado por terceiros desde que respeitadas as regras e boas práticas internacionalmente aceites, no que concerne aos direitos de autor e direitos conexos.

Assim, o presente trabalho pode ser utilizado nos termos previstos na licença abaixo indicada.

Caso o utilizador necessite de permissão para poder fazer um uso do trabalho em condições não previstas no licenciamento indicado, deverá contactar o autor, através do RepositóriUM da Universidade do Minho.

Licença concedida aos utilizadores deste trabalho

Creative Commons Atribuição-NãoComercial-CompartilhaIgual 4.0 Internacional
CC BY-NC-SA 4.0

https://creativecommons.org/licenses/by-nc-sa/4.0/deed.pt

Agradecimentos

Em primeiro lugar gostaria de agradecer ao Professor Doutor Paulo Jorge Freitas de Oliveira Novais pela oportunidade de trabalhar neste tema e por todos os ensinamentos nesta reta final do meu percurso académico.

À Ana Paula Pinto da Silva pela disponibilidade única, por toda a paciência e acompanhamento dedicado e pelas muitas aprendizagens que pude realizar e que me permitiram terminar esta dissertação. Por tudo isto, um muito obrigado.

Aos meus amigos e colegas que conheci na universidade, todos me ensinaram algo. Ao Braga, João, Henrique, Moreira, Ferreira, Milhazes, especialmente, por me acompanharem mais de perto, sem eles este percurso académico não teria sido tão fácil e bonito.

Aos meus amigos da minha terra natal, pela companhia diária e pelo incentivo.

À Rita Pinho, por nunca me deixar ir abaixo, pela força que me transmitiu diariamente em dias menos bons, pela motivação e pela paciência para me ouvir durante este ano.

À minha família pelo apoio e carinho incondicional. Ao meu irmão mais novo por me alegrar diariamente. Ao meu irmão mais velho por ser um exemplo e um ídolo para mim.

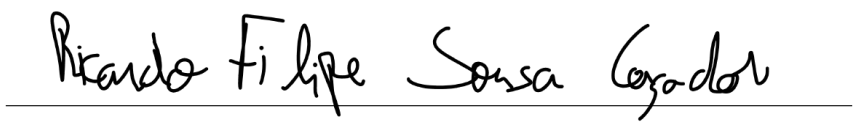
Quero dedicar esta dissertação ao meu pai e à minha mãe por me terem proporcionado todas as condições ideais para seguir e realizar os meus sonhos, por me apoiarem sempre e incondicionalmente independentemente das minhas escolhas e por me fazerem muito feliz. Espero um dia poder retribuir tudo o que fizeram por mim. Obrigado por tudo!

DECLARAÇÃO DE INTEGRIDADE

Declaro ter atuado com integridade na elaboração do presente trabalho académico e confirmo que não recorri à prática de plágio nem a qualquer forma de utilização indevida ou falsificação de informações ou resultados em nenhuma das etapas conducente à sua elaboração.

Mais declaro que conheço e que respeitei o Código de Conduta Ética da Universidade do Minho.

Braga, 27 de Dezembro de 2021



(Ricardo Filipe Sousa Caçador)

Resumo

A proximidade entre a Informática e a Saúde é cada vez maior a cada dia que passa. Nos dias que correm é comum os hospitais guardarem eletronicamente todo o historial e relatórios clínicos dos utentes.

O armazenamento digital destes dados traz vantagens aos sistemas de saúde como a acessibilidade, a otimização de recursos e redução de custos, a diminuição do erro médico e o auxílio nas tomadas de decisões. Grande parte desses dados está em formato de texto livre, ou seja, são dados não estruturados. Para os sistemas computacionais, este tipo de dados representa um maior desafio quer na análise, quer no seu processamento. Sendo que, para este tipo de informação ser processada automaticamente é necessário recorrer ao Processamento de Linguagem Natural, uma subárea da Inteligência Artificial. Tarefas como classificação ou reconhecimento de entidades em textos requerem quase sempre textos anotados.

O processo de anotação dos textos é demorado e pouco atrativo para o ser humano levando a que a quantidade disponível de dados anotados não seja em grande volume e consequentemente a que a aplicação de modelos de *Machine Learning* não seja a mais eficiente, resultado em problemas de *overfitting* e não generalizando como seria de desejar. Devido a isto, a procura por uma solução de anotação automática dos dados em massa é necessária e extremamente útil.

A principal contribuição desta dissertação é o desenvolvimento de uma aplicação para a anotação automática de informação clínica. Esta aplicação permitirá a anotação de grandes quantidades de dados de forma automática comparativamente a outras ferramentas e abordagens existentes.

Palavras-chave: anotação de textos automática, processamento de linguagem natural, extração de informação clínica, registo médico eletrónico

Abstract

The proximity between Informatics and Health is growing day by day. Nowadays, it is common for hospitals to store all the history and clinical data electronically.

The digital storage of these data brings advantages to health systems such as accessibility, optimization of resources and cost reduction, reduction of medical errors and help in decision-making. However, most of this data is in free-text format, that is, unstructured data. For computer systems, this type of data represents an enormous challenge both in analysis and processing. For this type of information to be processed automatically, it is necessary to resort to Natural Language Processing, a sub-area of Artificial Intelligence. Tasks such as classification or name entity recognition almost always require annotated text.

The process of annotating texts is time-consuming and unattractive for human beings, leading to the fact that the available amount of annotated data is not large. Consequently, the application of Machine Learning models is not the most efficient, resulting in overfitting problems and not generalizing as we would like. Due to this, the search for a solution of automatic annotation of clinical data is necessary and extremely useful.

The main contribution of this dissertation is the development of an application for the automatic annotation of clinical information. This application will allow the annotation of large amounts of data automatically compared to other existing tools and approaches.

Keywords: automatic annotation of text, natural language processing, clinical information extraction, electronic medical record

Índice

Lista de Figuras

Lista de Tabelas

Glossário

ICD — Terminologia desenvolvida pela OMS cujo objetivo é a promoção da comparação internacional da mortalidade através da codificação. Esta tem por base permitir o registo recorrente de dados relativos à mortalidade obtidos em diferentes países e em diferentes tempos. O número colocado a seguir à sigla refere-se à revisão ou modificação feita à terminologia. Em Portugal, foi adotada em 1989 e é utilizada amplamente pelo SNS desde então para efeitos de codificação clínica de altas hospitalares e também pelo sector privado. Com as constantes revisões e progressos, surge o ICD-10, desenvolvido pelo *National Center for Health Statistics* (NCHS), uma organização dos Estados Unidos da América. Esta revisão acrescentou à antiga versão 55 mil novos códigos de classificação de diagnósticos. Atualmente existem mais de 70 mil códigos na nomenclatura.

PubMed — O *PubMed* é um motor de busca clínico com acesso livre a uma base de dados com citações, artigos e resumos de investigação na área da Biomedicina. Possui quase 5 mil revistas publicadas nos Estados Unidos da América e está presente em quase 80 países de todo o mundo sendo a primeira publicação datada do ano 1966.

SNOMED-CT — Terminologia clínica internacional multilingue cuja língua oficial é o Inglês. A utilização desta terminologia permite registar informação em processos clínicos eletrónicos em diversos contextos desde sintomas de doenças até contextos mais sociais, passando por todo o tipo de relatórios. Esta terminologia é muito próxima da linguagem clínica natural. Esta está organizada por conceitos interrelacionáveis entre si o que permite o aumento da riqueza e, consequentemente, da qualidade dos dados. É importante referir que esta nomenclatura pertence a uma organização sem fins lucrativos, a SNOMED *International*, composta por 27 países, sendo Portugal um dos países pertencentes, desde Janeiro de 2014, através da Serviços Partilhados do Ministério da Saúde (SPMS).

SNS — O Serviço Nacional de Saúde é o serviço através do qual o Estado Português assegura o direito à saude a todos os cidadãos portugueses. Foi criado em 1979 e atualmente é constituído por 212 hospitais e 363 centros de saúde

Acrónimos

ACSS	Administração Central do Serviço de Saúde
ARS	Administração Regional de Saúde
BERT	*Bidirectional Encoder Representations from Transformers*
CBOG	*Continuous Bag-of-Words*
CNN	*Conventional Neural Networks*
GPU	*Graphic Processor Unit*
HIPAA	*Health Insurance Portability and Accountability Act*
i2b2	*Informatics for Integrating Biology & the Bedside*
IA	Inteligência Artificial
ICD	*International Classification of Diseases*
IGIF	Instituto de Gestão Informática e Financeira da Saúde
MALLET	*Machine Learning Language Toolkit*
MIMIC	*Medical Information Mart for Intensive Care*
ML	*Machine Learning*
NCHS	*National Center for Health Statistics*
NER	*Named Entity Recognition*
OMS	Organização Mundial de Saúde

PLN	Processamento de Linguagem Natural
RME	Registo Médico Eletrónico
RNN	*Recurrent Neural Networks*
SIARS	Sistema de Informação da Administração Regional de Saúde
SINUS	Sistema de Informação para as Unidades de Saúde
SNOMED-CT	*Systematized Nomenclature of Medicine Clinical Terms*
SONHO	Sistema Integrado de Informação Hospitalar
SPMS	Serviços Partilhados do Ministério da Saúde
SVM	*Support Vector Machine*
SWOT	*Strenghts Weaknesses Opportunities Threats*
TF-IDF	*Term Frequency-Inverse Document Frequency*
TPU	*Tensor Processor Unit*
UCI	Unidade de Cuidados Intensivos

Capítulo 1

Introdução

1.1 Motivação

A Inteligência Artificial (IA) define-se como o ramo da ciência da computação que desenvolve sistemas que simulam capacidades do ser humano, como raciocinar e tomar decisões. O ser humano tem algumas habilidades que nenhum outro animal possui e, como tal, desenvolver ou simular estas capacidades em diferentes sistemas é uma das maiores ambições da IA. A Saúde é um dos campos mais propícios à aplicação da IA, através das suas inúmeras técnicas e métodos cujas aplicações visam auxiliar tanto os pacientes como os profissionais de saúde, nas suas tarefas diárias ou em qualquer tipo de cuidado de saúde tendo sempre como objetivo aumentar a qualidade da área [1] [2]. A capacidade de detetar padrões e analisar de grandes conjuntos de dados, permitem auxiliar e dar suporte no processo de tomada de decisão dos profissionais de saúde, utilizando algoritmos de captação de conhecimento que por vezes poderão não ser percetíveis, mesmo a profissionais com mais experiência [3] [4].

A adoção e utilização, cada vez mais frequente, do Registo Médico Eletrónico (RME) por parte das instituições de saúde possibilita às mesmas guardar enormes conjuntos de dados, em larga escala, e a troca fácil e eficiente entre as mesmas de informações sobre os pacientes, como relatórios de altas e historiais clínicos. Por exemplo, em 2010, segundo a Administração Regional de Saúde (ARS), apenas 3 Centros de Saúde não se encontravam informatizados e todos as Unidades de Saúde Familiares estavam informatizadas [5]. Atualmente, pode-se dizer que praticamente todos os registos clínicos são informatizados e guardados em bases de dados eletrónicas estando disponíveis e acessíveis por todas as unidades de saúde do SNS e privados. Em Portugal o RME, lançado em Junho de 2012, já conta com mais de 530 instituições, incluindo todos os Centros de Saúde e todos os Hospitais do SNS [6].

A constante evolução, a ritmo elevado, da informatização da área da Saúde solicita, cada vez mais, o

avanço em paralelo da área da Informática, para solucionar os problemas dos dados não estruturados, da automatização de tarefas e anotação automática [7]. A inexistência de uma aplicação para anotação destes registos clínicos, direcionada para os profissionais de saúde, e para as próprias instituições, não possibilita o aproveitamento máximo desta informatização que tem ocorrido no presente e irá continuar a acontecer no futuro [8] [9].

Apesar de tudo, estes avanços tecnológicos na área da Saúde deram a origem a que cada vez mais cientistas, organizações e empresas caminhassem na direção do desenvolvimento de ferramentas e modelos de anotação de informação clínica para tarefas de previsão, reconhecimento de termos médicos e correção de erros em notas clínicas. Num estudo feito em 2014, mais de 100 milhões de notas clínicas eram revistas todos os anos, nos Estados Unidos da América [10]. Algumas em formato de texto livre, outras em registos clínicos estruturados.

A aplicação desenvolvida nesta dissertação utiliza então as tecnologias atuais de desenvolvimento de *software* juntamente com os avanços efetuados relativamente à Saúde na área do PLN.

1.2 O Uso do Registo Clínico Eletrónico em Portugal

O registo clínico é um conjunto ordenado de documentos que contêm todos os dados, tanto médicos como administrativos, que foram recolhidos a um utente. Estes contêm dados como: historiais clínicos, exames realizados, diagnósticos, tratamentos, prescrições, testes laboratoriais, entre outros [11].

O processo clínico envolve uma série de etapas, começando pela observação dos dados recolhidos do paciente que inclui vários tipos de dados narrativos e alguns numéricos como temperatura corporal, tensão arterial e idade. Estes dados geram vários tipos de informação que, com base no conhecimento do profissional de saúde, originam diagnósticos e interpretações clínicas (Figura 1).

Figura 1: Organização de um processo clínico clássico [12].

Os registos clínicos eletrónicos vão muito além da informatização de informação que outrora era escrita em papel. Estes sistemas trazem inúmeras vantagens a todos os intervenientes no sistema de saúde, principalmente a pacientes e a profissionais de saúde (Figura 2):

- Auxílio nas tomadas de decisões dos profissionais de saúde;
- Gestão e planeamento dos cuidados de saúde;
- Auxílio na investigação médica;
- Auxílio na educação médica.

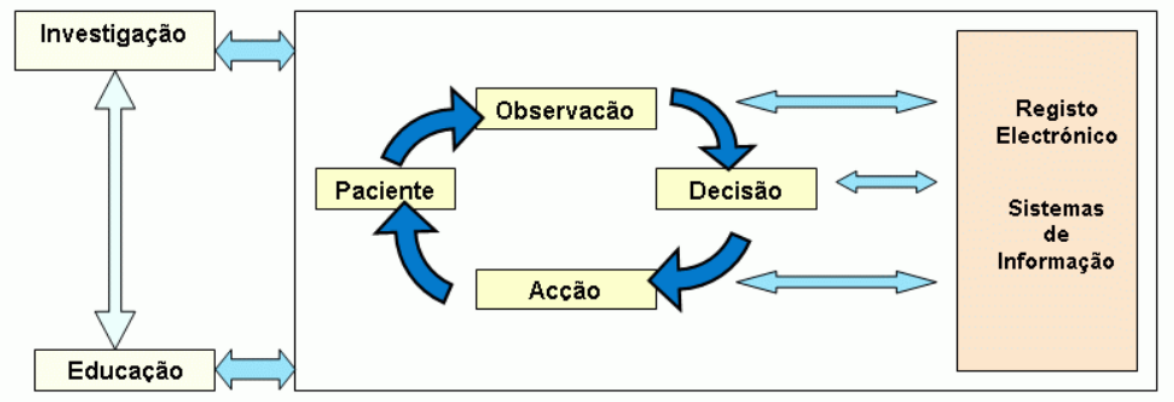


Figura 2: Organização e relação do registo clínico eletrónico com os sistemas de informação [12].

Em Portugal, a primeira utilização do Registo Eletrónico surge em 1994, com a aplicação Sistema Integrado de Informação Hospitalar (SONHO). Este software desenvolvido pelo Instituto de Gestão Informática e Financeira da Saúde (IGIF), organismo que passado 2 anos foi extinto, sendo substituído pela Administração Central do Serviço de Saúde (ACSS), que criou o Sistema de Informação para as Unidades de Saúde (SINUS) e o cartão de utente, este último que se tornou obrigatório desde 2000 [13].

Atualmente, segundo o coordenador dos Projetos Internacionais dos SPMS, Diogo Martins, o SNS comporta 54 hospitais e 359 outras unidades de saúde primárias para assistir cerca de 10 milhões de portugueses. O coordenador afirma numa entrevista que aproximadamente 90% das entidades de saúde utiliza soluções informáticas desenvolvidas pelos SPMS, tanto para cuidados de saúde como para problemas administrativos [14].

Relativamente a estas soluções informáticas, aproximadamente 98% do sector hospital, incluindo tanto público como privado, e cuidados primários, já utiliza a prescrição médica eletrónica. O SPMS desenvolveu portais para os registos clínicos eletrónicos que são constituídos por 2 áreas: a área profissional (dos clínicos) e a área pessoal dos pacientes. A informação do paciente, como por exemplo os dados imunológicos, pode ser utilizada pelo sector da saúde público e privado, hospitais, farmácias e até centros de vacinas mundiais, algo particularmente útil nos meses de pandemia que atravessámos. Quanto à área dos profissionais de saúde, uma das grandes vantagens deste sistema é que todos os profissionais que trabalhem no SNS têm acesso a todos os registos clínicos eletrónicos partilhados entre todas as unidades de saúde. Como tal, todos os hospitais que utilizarem o mesmo software de administração podem aceder a todos os registos eletrónicos armazenados neste sistema e realizarem comunicações entre si [14].

1.3 Objetivos

O principal objetivo desta dissertação passa por desenvolver uma aplicação para a anotação automática de informação clínica que permita auxiliar os profissionais de saúde no exercício da sua função [15] [16] [17].

Para realização deste projeto de dissertação foram delineadas as seguintes tarefas/metas:

1. Pesquisa e estudo das abordagens existentes (artigos científicos) acerca do PLN no âmbito de historiais clínicos/ RME;
2. Análise das ferramentas já existentes de anotação e das suas limitações;
3. Desenvolvimento de uma aplicação de anotação automática de informação clínica;
4. Exploração da aplicação desenvolvida e avaliação da mesma;
5. Análise dos modelos da aplicação e dos resultados dos testes.

1.4 Metodologia

Ao longo do desenvolvimento deste projeto será adotada uma metodologia de ação-investigação, pretendendo-se formular uma hipótese que responda a um dado desafio. Após o levantamento de toda a informação relevante, proceder-se-á à elaboração de uma proposta para resolução do problema. Após essa etapa, de modo a avaliar os resultados obtidos, serão efetuadas as respetivas conclusões. Assim, esta dissertação desenvolveu-se da seguinte forma:

- **Primeira etapa:** definição do problema e descrição das suas características.
- **Segunda etapa:** revisão do estado da arte, tecnologias e avanços relacionados com a anotação automática de informação clínica.
- **Terceira etapa:** conceptualização e desenvolvimento de um modelo com o objetivo de solucionar de maneira inédita e válida o problema, tendo como base a informação recolhida na etapa anterior.
- **Quarta etapa:** experimentação e implementação da solução obtida.
- **Quinta etapa:** análise e avaliação dos resultados obtidos durante o projeto de investigação.

1.5 Estrutura do Documento

Esta dissertação é composta por 7 capítulos, estruturados da seguinte forma:

- **Introdução:** no primeiro capítulo são introduzidos todos os conceitos importantes para a compreensão total do trabalho. Neste, é feita uma breve descrição da importância deste trabalho. São apresentados a motivação, os objetivos, a metodologia da dissertação e a sua estrutura;
- **Estado da Arte:** o segundo capítulo desta dissertação apresenta as ferramentas já existentes para anotação de informação clínica bem como o funcionamento das mesmas, os seus pontos fortes e fracos;
- **Dados Clínicos, Modelos e Arquitetura:** neste capítulo é feita a apresentação da base de dados e das notas clínicas utilizadas nesta dissertação. É feito um pequeno resumo geral sobre o estado dos modelos de PLN, mais especificamente, dos modelos pré-treinados; além disso é descrito o desenho da aplicação começando pela recolha dos requisitos até à escolha dos modelos;
- **Implementação:** no quinto capítulo da dissertação é descrito todo o processo de desenvolvimento da aplicação, desde a organização da aplicação em si, passando pela interface e finalizando com os *use cases* de funcionamento da mesma;

- **Prova de Conceito e Avaliação:** no penúltimo capítulo é feita a avaliação dos modelos utilizados na aplicação. Primeiramente é feita uma prova de conceito e depois são descritos os testes realizados aos modelos utilizados na aplicação assim como a metodologia utilizada nos mesmos;
- **Conclusões e Trabalho Futuro:** finalizando o trabalho, o último capítulo serve de síntese do trabalho alcançado. Esta conclusão é completada perspetivando o trabalho futuro que poderá ser realizado.

Capítulo 2

Estado da Arte

Neste capítulo será discutido e analisado o trabalho e as ferramentas existentes que se enquadrem no contexto desta dissertação.

2.1 Sumarização Automática

Feblowitz J, Wright A, Singh H et al. (2011) [18]

Em 2011, foi publicado um trabalho cujo objetivo era a idealização de um modelo conceptual para uma framework de sumarização automática de informação clínica. Segundo estes, os registos clínicos podem ser divididos em 3 grandes categorias interrelacionadas entre si:

- *Source-oriented view*: esta é a categoria que deriva da sumarização tradicional em papel na qual a informação é preenchida em diferentes categorias para facilitar o acesso a documentos. A informação é organizada de acordo com a sua fonte;
- *Time-oriented view*: este tipo de vista organiza a informação com base na altura da sua recolha e apresenta todos os dados organizados cronologicamente;
- *Concept-oriented*: por fim, nesta categoria organizada por conceitos toda a informação é tratada segundo conceitos clínicos como problemas médicos ou órgãos do corpo humano. Esta *view* acelera o processo de procura de informação e aumentam a qualidade da tomada de decisão médica.

Os registos clínicos podem ser caracterizados como concretos, quando a informação clínica é condensada sem qualquer alteração, ou abstratos, quando aos registos clínicos é aplicada informação de

contexto para criar um sumário clínico mais sofisticado. Os sumários clínicos podem também ser considerados pobres ou ricos consoante o valor da sua informação. Por exemplo, formas simples de registos clínicos como gráficos de sinais vitais são considerados registos clínicos concretos pobres.

Os autores deste trabalho, em 2011, analisaram a literatura sobre sumarização clínica e ferramentas de anotação automática e identificaram conceitos importantes comuns em todos.

1. **Os formatos dos registos clínicos são heterogéneos:** existe uma grande variedade de métodos para o registo clínico pelas várias instituições de saúde como: trocas verbais, notas escritas à mão, documentos de Word e/ou outros registos gerados por computadores;

2. **Falta de instrução para criação de registos clínicos:** à data da escrita não havia nenhum tipo de formação ou instrução consistente para melhorar o registo deste tipo de documentos;

3. **Tentativas de standardização dos registos clínicos:** foram identificadas pelos autores várias tentativas e esforços para standardizar o formato e apresentação dos registos clínicos de forma a facilitar os processos de sumarização automática e a qualidade dos registos clínicos tornando-os mais completos;

4. **Limitação dos estudos sobre anotação clínica:** entre 1977 e 2005 os autores citam uma falta de investigação de alta qualidade neste tema e que a maioria da pesquisa na anotação se focava apenas na produção de registos clínicos apenas de um ou alguns documentos de texto.

No ano de escrita deste trabalho e baseado nas tendências encontradas à data da pesquisa de outros trabalhos e abordagens foram identificados 4 objetivos que um modelo conceptual de sumarização devia atingir:

1. Providenciar uma *framework* comum aplicável a diferentes tipos de registos clínicos;

2. Análise de registos clínicos gerados por humanos e/ou computadores;

3. Facilitar a standardização e/ou automação dos registos clínicos;

4. Encorajar a pesquisa futuro na sumarização clínica.

Com base neste trabalho, na análise feita e na identificação destes conceitos desenvolveram uma proposta para um modelo de sumarização ao qual deram o nome de modelo AORTIS composto por 5 passos fundamentais:

- **Agregação:** recolha dos vários dados clínicos das mais diversas fontes;

- **Organização:** nesta etapa é feita a estruturação dos dados sem ser feita quaisquer alteração nos mesmos. Os dados são agrupados e ordenados;

- **Redução/Transformação:** nestas 2 etapas deve ser resolvido o problema do excesso de informação desnecessária. Para tal, é feita a redução, processo de eliminação de informação da base de dados que não altera em nada a qualidade da mesma. O processo de transformação contempla a alteração de como os dados estão apresentados ou a densidade dos mesmos de forma a facilitar a compreensão;

- **Interpretação:** é a análise baseada em contexto médico dos dados clínicos que temos. Nesta etapa são necessários conhecimentos médicos pois estes permitem ao utilizador selecionar os dados que são relevantes em contexto;

- **Síntese:** a fase final é a combinação de elementos de dados com conhecimento e interpretação médica de forma a sugerir uma decisão médica. Esta é a etapa mais valorizada da anotação clínica pois é nesta que se torna possível a criação de vistas orientadas em contexto (referidas anteriormente).

Segundo este grupo de investigadores uma ferramenta de anotação automática que seguisse a proposta de sumarização dos mesmos deveria apresentar resultados idênticos aos presentes nas seguintes imagens (Figura 3 e 4):

John Q. Smith – 375 Plantation Rd. Luling, TX W: 713-985-4215 Insurance: BC/BS TX
67yr white male 5'-9" 195 lbs (↓4 lbs in 12 mo.) BMI-28.8

Diabetes Risk Management Summary

Glycemic Control: **Type 2 DM** (dx: 10/1/09): HbA1c-7.0% (10/01/10) (↓3.0% in 12 mo.) on metformin (1000 mg BID).
Glycemic control is acceptable according to ADA guidelines.

Lipid Control: **Hyperlipidemia** (dx: 10/01/09): Total cholesterol-250 mg/dL, HDL 40 mg/dL, LDL 175 mg/dL (10/1/10) (↑ from 180/60/125 4 mo. ago) on simvastatin (20 mg QD)
ATP III guidelines recommend adjusting dosage.

Blood Pressure Control: **Hypertension** (dx: 10/01/09): BP-135/90 (today) (↓from 150/105 in 12 mo.) on Hydrochlorothiazide (25 mg QD)
JNC VII guidelines recommend adding a medication.

Visit History: Clinic – Urgent Follow-up (6/15/10); ED – Hospital – chest pain (6/1/10); Clinic – Well Visit (2/1/09); Clinic – Physical (10/1/09)

Figura 3: Exemplo de sumarização clínica automática [18].

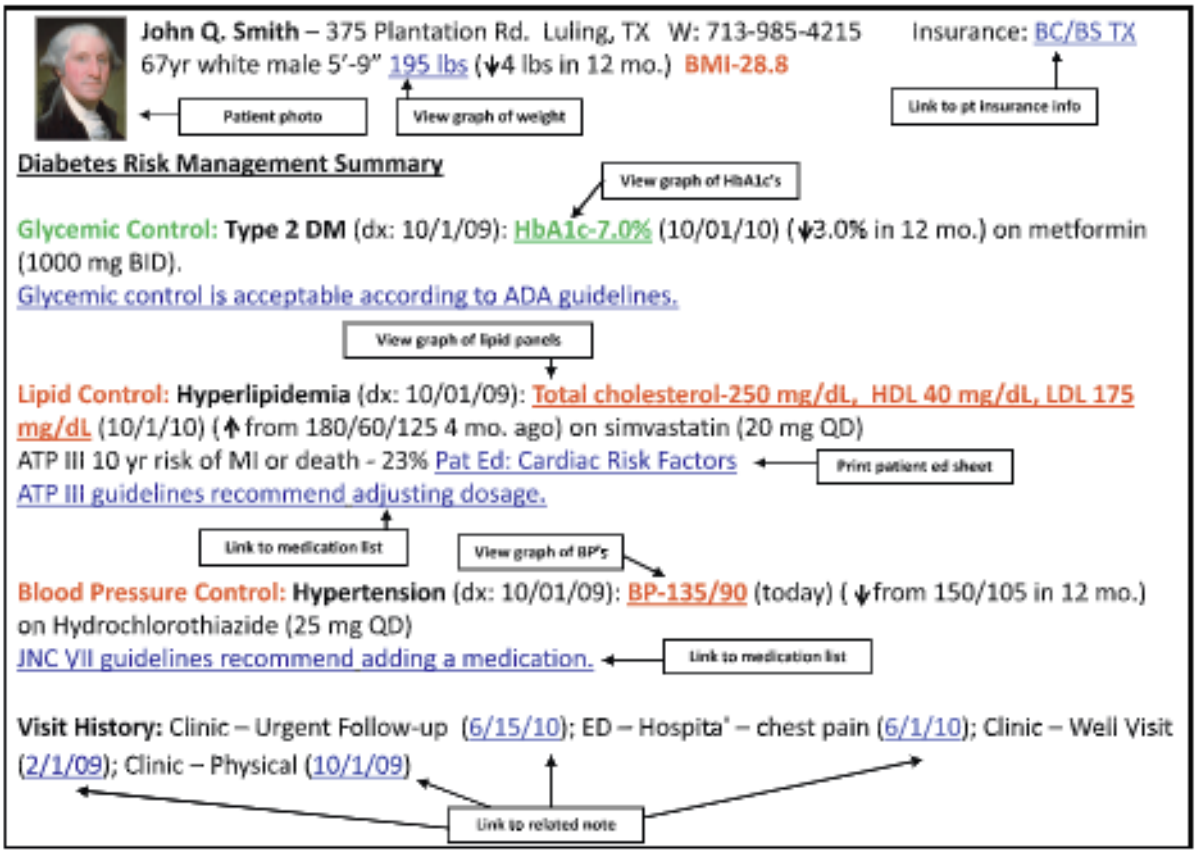


Figura 4: Exemplo de um resumo anotado [18].

Relativamente às etapas definidas no modelo AORTIS os autores resumiram os principais desafios e potenciais soluções para os mesmos (Figura 5):

	Challenges and problems	Potential solutions and future research directions
Aggregation	Incomplete electronic data	Increase data capture (improve user and system interfaces, device connectivity, voice recognition, data entry)
	Records distributed across multiple healthcare and information systems	Health information exchanges (HIEs)
	No unique patient identifier or central patient index	Community-wide master patient index (MPI), statistical matching algorithms
	Health information portability and accountability act (HIPAA)	Business-associate agreements
Organization	Lack of controlled, structured, and coded data	Template data entry, natural language processing, development and use of standard terminologies
	Many jobs require task-specific organization techniques	Expanded clinical knowledge bases
	Dealing with temporal data and logic	Increased research in temporal systems in computer science and informatics (e.g. TSQL) [70,71]
Reduction	Reduce data while preserving meaning	New statistical methods
	No methods for prioritizing data	Approaches for prioritizing data elements to display
	Task-specificity of identifying relevant information	Improved understanding of relationships between different clinical data elements (relevance) [66]
	Inaccurate or irrelevant data	Robust method of distinguishing between clinically-significant outliers and noise/bad data
	Mismatch between clinical and statistical significance	New statistical methods (communication)
Transformation	Limited understanding of optimal data transformation for clinical reasoning	Additional research on workflow and clinical reasoning
	Limited EHR functionality to display data in various forms	Increased capabilities of EHR systems
	Task-specific understanding of trend	Investigation of task-specific needs and development of knowledge bases
	Superimposition of secular trends (confuses understanding of clinical trend)	More robust knowledge and better methods for separate trends (e.g. time-series analysis and Kolman filters)
Interpretation	Task- and condition-specificity for interpretations	Create a taxonomy of clinical tasks, conduct additional research on cognitive and workflow needs of different clinical tasks, develop new condition-specific knowledge bases to help interpretation
	Lack of understanding of clinician cognition	Additional research needed [72,73]
	Heterogeneity of mental models of clinical processes	Study of cognition, standardization of treatment guidelines
Synthesis	High dependency on prior steps of the model and propagation of errors or missing data	New research and more advanced clinical tools to support aggregation, organization, reduction, transformation and interpretation
	Limited understanding of human pathophysiology	Research on mechanisms of diseases
	Lack of adequate computable clinical knowledge bases	Research to design, develop, implement and test new standardized clinical knowledge structures

Figura 5: Principais desafios e potenciais soluções para o modelo conceptual de sumarização clínica [18].

Esta tabela permitiu perceber que muitos dos problemas que existiam em 2011, continuam a existir à data de hoje.

2.2 Representação de Texto e Modelos de Classificação

Berndorfer S and Henriksson A (2017) [19]

O objetivo deste estudo passou por comparar e perceber as diferenças em termos de desempenho dos diferentes tipos de representação de textos e modelos de classificação. Através desta análise foi possível perceber a eficácia de cada combinação de estratégia utilizada nos modelos e aplicada nos registos clínicos do *MIMIC-III* que utilizavam a nomenclatura *ICD-9*. Esta base de dados armazena grandes quantidades de registos clínicos, não identificados, de pacientes admitidos nas unidades de saúde do *Beth Israel Deaconess Medical Centre* em Boston, Massachusetts nos Estados Unidos da América. Esta base de dados com mais de uma década integra vários tipos de registos clínicos:

- **Faturação:** dados codificados usados primariamente para faturação e administração da instituição clínica;

- **Informação descritiva:** informação demográfica, horários de admissões, altas e mortes hospitalares;
- **Dicionários:** tabelas com códigos e identificadores e as respetivas definições;
- **Intervenções clínicas:** procedimentos como diálises e pequenos curativos;
- **Relatórios laboratoriais:** exames laboratoriais como análises ao sangue, à urina e outros testes biológicos;
- **Receitas médicas:** registos das receitas médicas e administrações ao paciente;
- **Notas clínicas:** notas em texto livre escritas pelos clínicos;
- **Relatórios fisiológicos:** registos de sinais vitais, verificados por enfermeiras (batimentos cardíacos, pressão arterial e respiração);
- **Relatórios de exames:** registos livres relativos a vários tipos de exames médicos.

Para o algoritmo de aprendizagem foram utilizados 2 tipo de representação do texto e foram analisadas ambas as performances:

- *Shallow:* esta representação descreve cada registo clínico como *bag-of-words*. Neste tipo de representação, um texto é considerado um saco de palavras com uma frequência distribuída em cada palavra utilizando os valores *Term Frequency-Inverse Document Frequency* (TF-IDF) nos dados de treino. O valor TF-IDF é uma medida estatística cujo intuito é indicar a importância de uma palavra num registo em relação a um coleção de registos [20].
- *Deep:* nesta abordagem, a representação é feita caracterizando cada registo clínico como a soma dos valores *TF-IDF* dos vetores semânticos que foram treinados segundo um modelo de *Continuous Bag-of-Words (CBOG)* utilizando a técnica de processamento de linguagem natural *Word2Vec*. Este modelo tenta prever uma palavra baseando-se nas palavras que rodeiam o contexto onde estará inserida. O modelo *Word2Vec* é um modelo de aprendizagem profunda que tem como objetivo gerar vetores semânticos que representam palavras de acordo com o contexto de captura e similaridade semântica [21] [22].

Depois dos modelos estarem treinados estabeleceram-se várias estratégias de combinação dos mesmos consoante esta ser feita antes ou depois da aprendizagem:

- *Fusion*: concatenação de *features* dos 2 tipos de representação abordados antes de partir para aprendizagem do modelo;

- *Select One*: nesta abordagem posterior à aprendizagem dos 2 modelos, é feita uma escolha de uma das representações baseando-se na performance dos 2 modelos;
- *Union* e *Intersection*: estratégia de união ou interseção das 2 previsões;
- *Probability Averaging*: nesta estratégia opta-se pela média do peso das probabilidades das classes produzidas pelos modelos. Os pesos são determinados pela performance de previsão observada em cada intervalo de frequência.

Após a combinação dos 2 modelos os autores utilizaram 2 modelos de classificação: o SVM normal e um modelo SVM hierárquico com base na hierarquia da nomenclatura ICD-9.

O modelo SVM é um modelo de aprendizagem supervisionada que analisam dados e reconhecem padrões utilizando classificação e métodos de análise de regressão. Neste modelo, para cada entrada de conjuntos dados é feita uma previsão para qual das duas possíveis classes esta entrada fará parte.

Estas técnicas foram aplicadas na base de dados *MIMIC-III*, mais especificamente, em todos os registos clínicos que utilizavam codificação ICD-9. Os códigos com menos de 50 ocorrências foram excluídos da base de dados. Esta remoção resultou em 59 531 registos clínicos.

Algumas considerações sobre o *dataset* final:

- A média de palavras por registo clínico era de 742 palavras;
- Foram encontrados 1 301 códigos ICD-9 distintos;
- A média de códigos por registo clínico foi de 10.66.

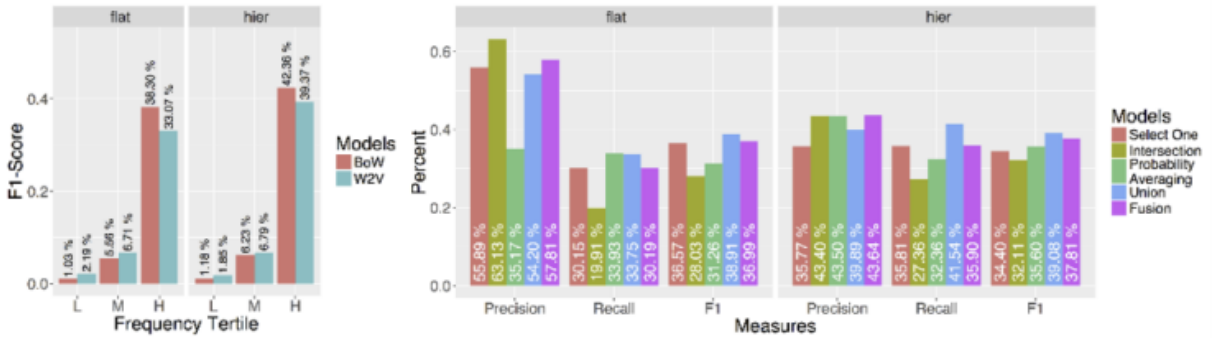


Figura 6: Análise aos modelos e às combinações dos mesmo [19].

A análise aos resultados da aplicação dos modelos (Figura 6) permitiu concluir que a eficiência relativa dos 2 tipos de representação do texto dependem da frequência de códigos de diagnóstico que estamos a considerar. Para códigos com frequência baixa-média, o modelo de representação *deep* teve melhores resultados principalmente devido à falta de exemplos para treinar o modelo. Relativamente às estratégias

de combinação, a utilização de uma união dos 2 modelos, apesar de simples, foi a que teve melhores resultados. Em geral um modelo combinado foi sempre melhor que um único modelo simples.

2.3 Anotação Automática no Diagnóstico de uma Patologia

Horng S, Sontag D, Halpern Y et al. (2017) [23]

O objetivo deste trabalho passou por demonstrar em como a utilização de texto livre juntamente com sinais vitais e informação demográfica ajudaria na identificação de pacientes com suspeita de sepse. A sepse é uma condição que pode ser fatal num ser humano e surge quando a resposta do corpo a uma infeção danifica os tecidos e órgãos do mesmo. Os sintomas mais comuns desta condição são febre, ritmo cardíaco acelerado, frequência respiratória alta e confusão mental [24]. Este trabalhou focou-se então em identificar pacientes utilizando registos clínicos com codificação ICD-9 e texto livre para identificar a infeção durante a triagem nas urgências. Para a recolha de dados e processamento este estudo utilizou cerca de 275 mil visitas de urgência de um hospital.

Relativamente a este processo de recolha de dados, foram utilizadas 12 *features* presentes nos registos clínicos de urgência assim como os códigos ICD-9-CM de diagnóstico do registo eletrónico. Destes, os 10 primeiros foram considerados como vitais e os últimos 2 de texto livre. Para realizar a "tokenização" do texto livre das queixas do paciente e das notas do enfermeiro o processo passou por separar a pontuação das palavras iniciais e finais de cada frase e considerar qualquer sequência de símbolos separados por um espaço como um *token*. Após isso foi aplicada a deteção de bigramas, ou seja, termos como *chest pain* foram tornados palavras únicas através de hífens ficando *chest-pain*. Para termos como "sem febre" ou "sem arrepios" foi utilizado a deteção de negação passando a *tokens* do estilo *chest-pain_neg*. Para esta última alteração foi utilizado o algoritmo *NegEx*. Este algoritmo foi criado especificamente para detetar negações de condições clínicas [25]. Depois disto foi realizada a validação e normalização dos dados. Sinais vitais em falta foram preenchidos com valores fisiológicos considerados normais e todos os valores foram normalizados para preencher ao intervalo entre 0 e 1 (Figura 8).

Após os *datasets* estarem preparados foram construídos 3 modelos de treino, validação e teste. Estes modelos foram construídos utilizando *Machine Learning* (ML) através de SVM. Por motivos de comparação os modelos foram também treinados utilizando regressão logística, os algoritmos de classificação *Naive Bayes* e *random forests*. Foram treinados 4 modelos que são apresentados na figura seguinte (Figura 7):

	Vital Signs	Patient Demographics	Chief Complaint	Nursing Assessment
Vitals Model	X	X		
Chief Complaint Model	X	X	X	
Bag of Words Model	X	X	X	X
Topic Model	X	X	X	X

Figura 7: Modelos de previsão e *features* presentes [23].

Relativamente ao modelo *bag of words* e ao modelo com queixas dos pacientes estes incluem uma *feature* para cada palavra do vocabulário cujo valor é a frequência do termo definida pelo número de ocorrências dessa palavra no registo de um paciente.

No caso do último modelo, os textos livres dos registos clínicos foram processados aprendendo um conjunto de 500 tópicos principais usados frequentemente em pacientes de urgências. Depois, consoante as queixas do paciente e o texto da triagem é feita uma distribuição de tópicos. Assim para além dos sinais vitais, foram criadas *features* para cada tópico contendo a probabilidade de um paciente ter esse tópico. Para esta parte foi utilizado o software *open-source Machine Learning Language Toolkit (MALLET)* [26].

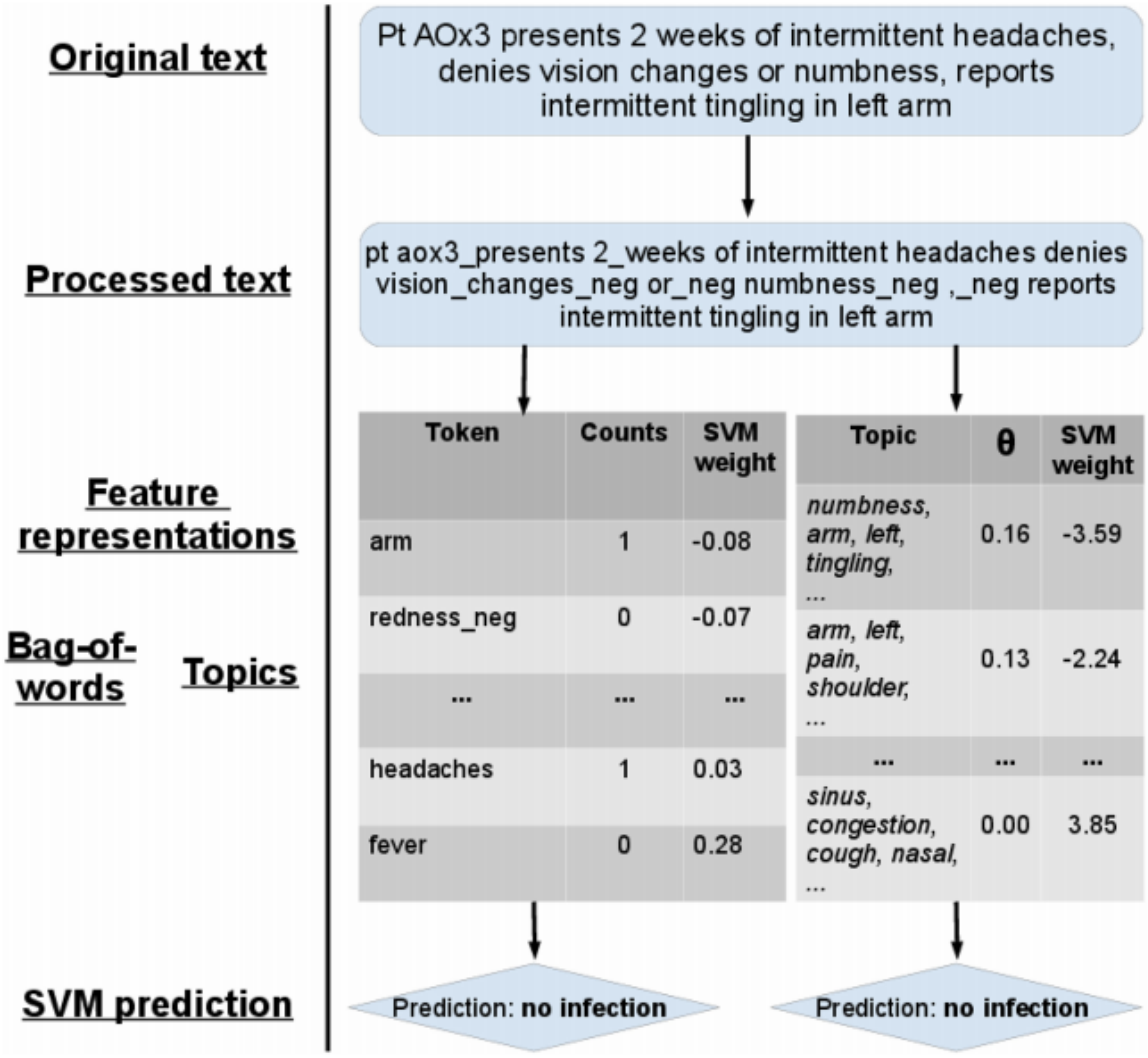


Figura 8: *Pipeline* de processamento de texto e previsão SVM [23].

Nas urgências, o tempo de decisão é crucial. A tomada da decisão clínica deve ser o mais rápida possível de forma a evitar que pacientes de risco sofram consequências graves com a demora na deteção de uma infeção. Por esta razão, métodos convencionais como administração de medicação, pedidos de exame ou registo de códigos de diagnóstico são decisões muito lentas e demoradas que este trabalho

pretende resolver. Relativamente ao objetivo deste trabalho, este foi atingido pois, comparativamente aos outros estudos de identificação da sepse em pacientes, a inclusão do texto livre no modelo revelou aumentar a qualidade deste obtendo um maior contexto das razões da visita do paciente às urgências e até, em alguns casos, sendo o texto livre o responsável por desqualificar a sepse como a causa do paciente estar ali.

2.4 Processamento de Linguagem Natural e uso de RSEs

Spasic I, Nenadic G (2020) [27]

Neste trabalho publicado recentemente os autores procuram evidências sistemáticas nas propriedades dos dados utilizados para aplicação de métodos de ML para PLN em dados clínicos. Neste artigo é também feita uma investigação quanto aos métodos de PLN suportados pelo ML e como é que estes são aplicados em termos práticos e contexto clínico.

Primeiro, começaram por definir perguntas de pesquisa às quais queriam obter resposta. O objetivo principal deste trabalho era responder a essas questões e, para isso, efetuaram uma pesquisa na interface *PubMed* (biblioteca de literatura biomédica da *MEDLINE*) da qual resultaram 110 estudos relevantes e extraíram a informação que necessitavam: tipo de dados usados com suporte a ML, métodos de PLN utilizados e aplicações práticas em contexto clínico feitas.

Um dos principais problemas observado foi a utilização de *datasets* mais pequenos mesmo com a possibilidade da utilização de *datasets* maiores. A razão para a muito fraca utilização dos dados disponíveis passa pelo facto de muitos destes necessitarem de intervenção humana para serem utilizados. Os algoritmos de ML supervisionados requerem que os dados de treino sejam anotados de forma a cobrirem os modelos matemáticos. A solução encontrada passou pela aplicação de algoritmos ativos com intervenção humana numa tentativa de aumentar a qualidade dos *datasets*.

Uma abordagem com algoritmos de aprendizagem supervisionada é mais conveniente quando as *labels* já estão definidas. Porém, os registos médicos eletrónicos possuem diferentes tipos de dados sejam eles estruturados (números, datas, códigos) ou não estruturados (texto livre ou imagens). Na revisão sistemática feita pelos autores percebeu-se que os *datasets* considerados grandes apenas eram utilizados, maioritariamente, quando os dados eram estruturados e anotados. Foram encontrados trabalhos de previsão utilizando para isso dados relativos a hospitalização, relatórios de altas, mortes hospitalares e readmissões. Quanto a diagnósticos, foram utilizados os códigos ICD para treino de modelos de previsão através de dados anteriores para identificação de pacientes em risco.

Outro dos problemas abordados foi a origem dos dados e chegou-se rapidamente à conclusão que as estruturas e os estilos dos dados podiam variar muito consoante as instituições de saúde de onde eram originários. Nesta revisão a maior parte dos estudos eram limitados às instituições de saúde dos autores,

o que levou a problemas de *overfitting*, ou seja, os dados utilizados não representavam uma grande escala e, assim, o modelo adaptava-se apenas aos dados utilizados e não generalizava de forma adequada. São raros os *datasets* com grandes quantidades de dados acessíveis de forma livre e fácil pela comunidade, sendo a MIMIC uma das poucas bases de dados deste género.

Este problema de acessibilidade às bases de dados resulta muitas vezes que um modelo aplicado numa instituição tenha resultados completamente diferentes se for aplicado noutra. Isto porque o formato, a estrutura e o estilo dos registos clínicos das bases de dados utilizadas podem variar muito entre unidades de saúde. Relativamente a este problema, o *dataset* da MIMIC desempenha um papel principal na pesquisa aberta do processamento de linguagem natural em contexto clínico por ser das poucas base de dados, até à data de escrita, de relevância extrema.

A maior parte das aplicações de PLN nos estudos revistos pelos autores focava-se no diagnóstico e previsão, aspectos fundamentais na Medicina.

Dos estudos analisados, relativamente a aplicações clínicas do PLN, a maior parte destes focava-se na classificação de texto que, naturalmente, tende para algoritmos supervisionados de ML. Os resultados eram utilizados maioritariamente para suporte ao prognóstico, manutenção de recursos e controlo.

Su Y, Chao CHung L et al. (2020)[28]

Em 2020, investigadores de Taiwan analisaram a utilização do RME, as suas vantagens e desvantagens e fizeram um estudo sobre a aplicação de técnicas de processamento de linguagem natural para resolução do problema da informação redundante e anotação de informação de registos clínicos. Estes concluíram que a identificação de nova informação em registos clínicos é possível e prática, e que isto proporciona aos clínicos uma maior facilidade na procura de informação-chave sobre o paciente e poupa carga de trabalho aos mesmos.

Após uma revisão sistemática para perceber o funcionamento e a utilização dos registos clínicos eletrónicos neste país algumas conclusões foram sendo retiradas. Cerca de 66% a 90% dos clínicos utiliza regularmente métodos de *copy-paste* e cerca de 80% utiliza esta técnica para documentação de pacientes. Esta técnica fez com que os registos eletrónicos ficassem sobrecarregados com muita informação igual e redundante. Segundo um questionário efetuado a utilizadores de RME estes admitiram que a facilidade de acesso a informação relevante de um paciente como o requisito principal procurado nos *softwares*. Assim, a maior parte dos estudos focou-se na alteração dos *designs* numa forma de tentar facilitar a visualização dessa mesma informação.

Ao longo dos anos muitos investigadores tentaram desenvolver aplicações à volta do PLN para sumarização e extração da informação necessária dos registos clínicos de pacientes de um RME. Este trabalho focou-se na investigação de 3 pontos importantes:

- A quantidade de nova informação vs informação redundante em registos clínicos;

- A precisão da identificação automática de nova informação em registos hospitalares;
- Qual o impacto da identificação de informação nova para a qualidade e carga de trabalho dos clínicos.

O ambiente de estudo foi o *Ditmanson Medical Foundation Chia-Yi Christian Hospital* localizado em Taiwan. Este hospital tem mais de 3000 trabalhadores com aproximadamente 47 000 admissões, 1 110 000 visitas ambulatórias 89 000 emergências por ano.

O método para analisar as notas clínicas de pacientes passou por escolher 10 pacientes dos cuidados intensivos. Estes pacientes tinham que ter estado hospitalizados por mais de 10 dias com condições mais complexas. Todas as notas clínicas foram revistas de forma a que estes pacientes não tivessem sido já tratados anteriormente pelos profissionais clínicos participantes no estudo. As identificações dos pacientes foram trocadas por um número de identificação do estudo para garantir confidencialidade.

A utilização de um modelo de linguagem estatística é útil na aplicação de técnicas de processamento de linguagem natural como reconhecimento de voz e categorização de texto. Neste estudo a identificação automática de informação foi feita utilizando um *bigram language model*. Um *n-gram model* é um tipo de linguagem que calcula a possibilidade de encontrar uma certa palavra baseada no facto de esta estar precedida de n-1 palavras numa frase. No caso do *bigram model* este estima a probabilidade de ocorrência de uma certa palavra quando é antecedida por outra palavra. Assim o método utilizado passou pelo tratamento do texto em termos de correção de gramática e aplicação deste modelo aos mesmos registos clínicos. Se um bigrama nunca tivesse sido identificado em registos clínicos antecedentes então seria considerado como nova informação e seria destacado.

Para avaliar a qualidade deste método foram utilizados 3 parâmetros: precisão, sensibilidade e o *F1 score*. A aplicação a cada um dos bigramas poderia ter 3 resultados: verdadeiro positivo, quando a informação nova fosse identifica tanto pela anotação automática como pela manual; falso positivo, quando a informação fosse identificada como nova pela anotação automática mas como repetida pela anotação manual e falso negativo quando a anotação automática considerasse essa frase como informação repetida mas fosse, de facto, nova informação.

Este estudo concluiu que a utilização de RME apesar das inúmeras vantagens, armazena muita informação redundante que pode, potencialmente, ser prejudicial à segurança de um paciente. Esta quantidade de informação redundante nos registos clínicos aliada a grandes cargas de trabalho dos profissionais de saúde pode comprometer a qualidade de serviço dos mesmos. Apesar disto, foi provado que a identificação automática de nova informação em registos clínicos é fazível e prática e permite que os utilizados dos registos médicos eletrónicos tenham um melhor conhecimento das condições dos pacientes e exerçam o seu trabalho diário mais levemente e de forma mais eficaz.

2.5 O Estado Atual das Ferramentas de Anotação de Texto

Ramos Javier (2021)[29]

Para perceber realmente o impacto desta dissertação no espaço temporal atual, era importante especificar o panorama de ferramentas de anotação de texto.

Neste artigo bem recente, escrito em Agosto de 2021, são enunciadas as principais ferramentas e é feita uma análise individual de cada uma comparando os seus pontos fortes e pontos fracos, bem como o seu funcionamento.

2.5.1 Doccano

O *Doccano* (Figura 9) é uma ferramenta *web open-source* de anotação. Esta ferramenta oferece ao utilizador bastante controlo sobre a mesma permitindo a modificação de código e oferecendo também a possibilidade de instalação no computador do utilizador.

Esta ferramenta, devido ao seu grau de dificuldade de utilização baixa é especialmente útil para utilizadores que não sejam programadores pois funciona de forma intuitiva e qualquer pessoa pode colaborar no processo. Além disso esta aplicação permite o trabalho em equipa permitindo um maior avanço da anotação paralelamente entre os membros.

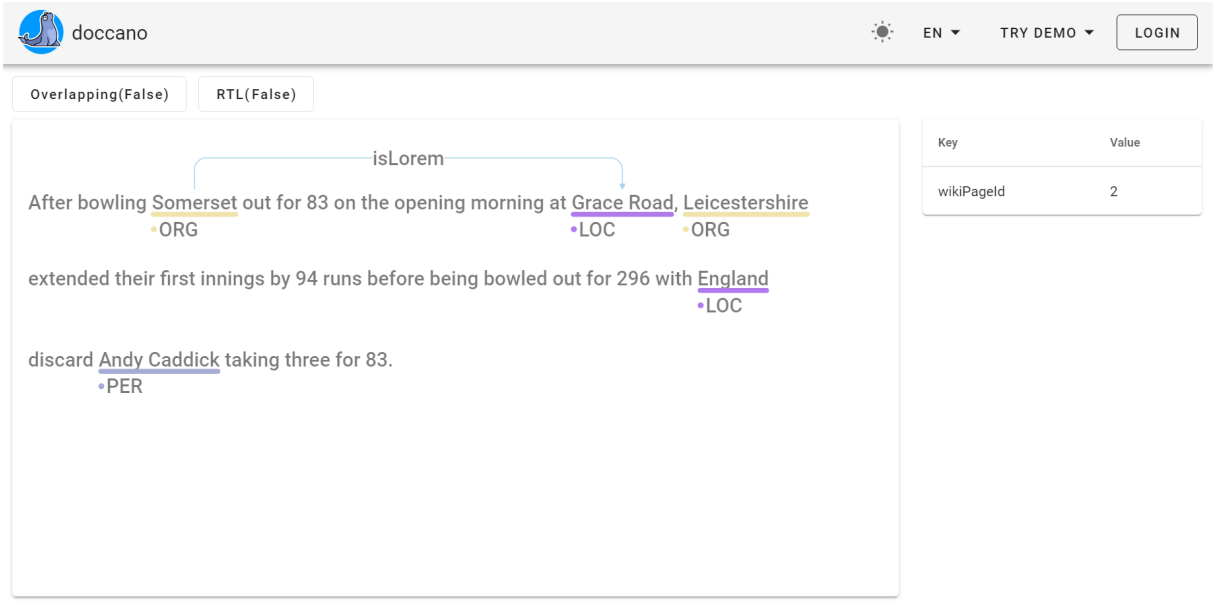


Figura 9: *Live demo* do *Doccano*.

2.5.2 Prodigy

O *Prodigy* (Figura 10) foi criado pela mesma equipa que desenvolveu a biblioteca *SpaCy*. Além de uma ferramenta de anotação de texto para a criação de treino e avaliação de dados para modelos, devido à sua integração com a biblioteca *SpaCy*, também pode ser utilizada para treinar os seus modelos. Assim, torna-se claro que o principal alvo desta ferramenta são cientistas com elevado conhecimento de *Python*.

Esta ferramenta pode-se considerar como semiautomática devido à aprendizagem ativa que possui. No início, o utilizador começa por anotar alguns textos e esta componente da ferramenta vai tentar aprender e começar a anotar o resto do *dataset* pelo utilizador. Assim, vai poupar algum trabalho ao utilizador que evitará perdas de tempo a anotar *samples* que não iriam melhorar a qualidade das previsões do modelo.

Do ponto de vista informático, esta é uma ferramenta bastante mais poderosa e que, com a sua integração com o *SpaCy* oferece aos programadores de PLN a possibilidade de implementação de ferramentas para processamento de linguagem de forma mais fácil e eficaz.

Apesar de todas estas vantagens, esta ferramenta não é *open-source*, tendo mesmo um custo associado para a obtenção de uma licença que permita a instalação da ferramenta no ambiente do utilizador.

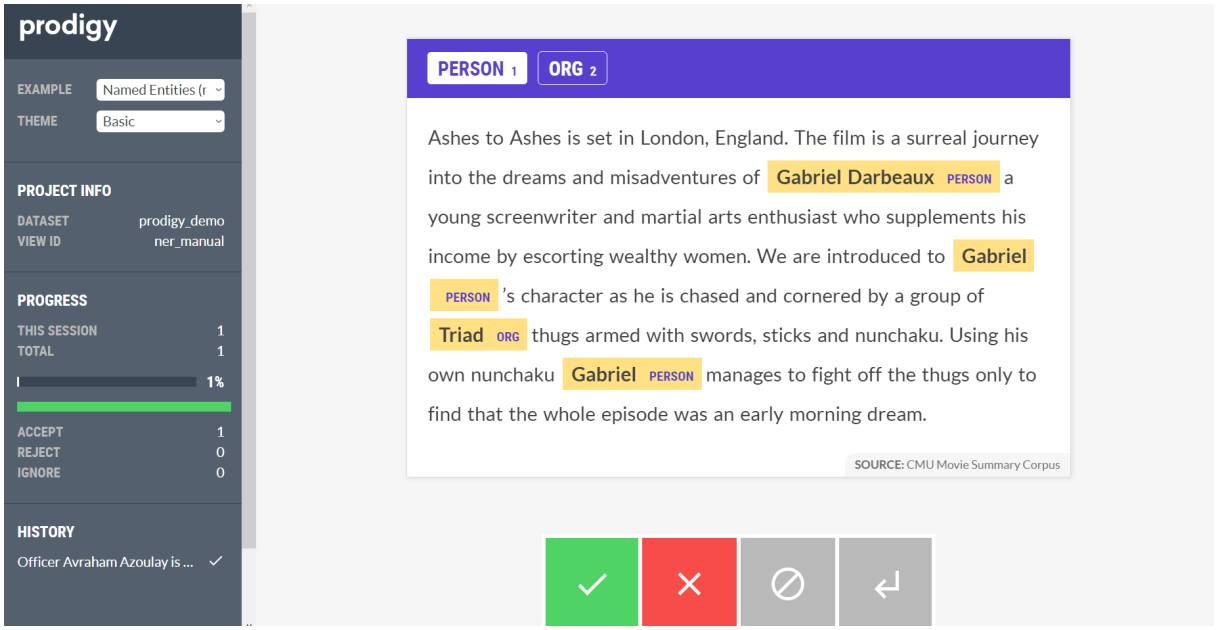


Figura 10: *Live demo* do *Prodigy*.

2.5.3 Outras Ferramentas

Além da existência destas 2 ferramentas com maior suporte e adesão da comunidade, é possível encontrar mais 2 com alguma utilidade, ainda que, para já, longe das 2 enunciadas anteriormente. É o caso do *TagTog* (Figura 11) e do *Bart* (Figura 12).

O *TagTog*, com mais avanços em relação ao *Bart*, já possui parcerias com algumas organizações e universidades importantes, apresenta anotação automática e manual de texto, proporciona a utilização

de modelos de ML semi-supervisionada e ainda disponibiliza cerca de 25 *datasets* de acesso livre à comunidade. É de salientar que a anotação automática não é disponibilizada de forma gratuita, ao contrário da anotação manual.

Quanto ao *Bart*, esta é uma ferramenta *web* designada particularmente para anotação de informação de forma estruturada e não de texto em formato livre daí não ser tão popular como as ferramentas enunciadas anteriormente. Esta apresenta funcionalidades como: um editor de anotação intuitivo, a integração com recursos externos de texto como o *Wikipédia*, a *Freebase* ou o *Open Biomedical Ontologies*. Além disso, como está desenvolvida sob tecnologias *web* não é necessário instalar nada para poder começar a anotar.

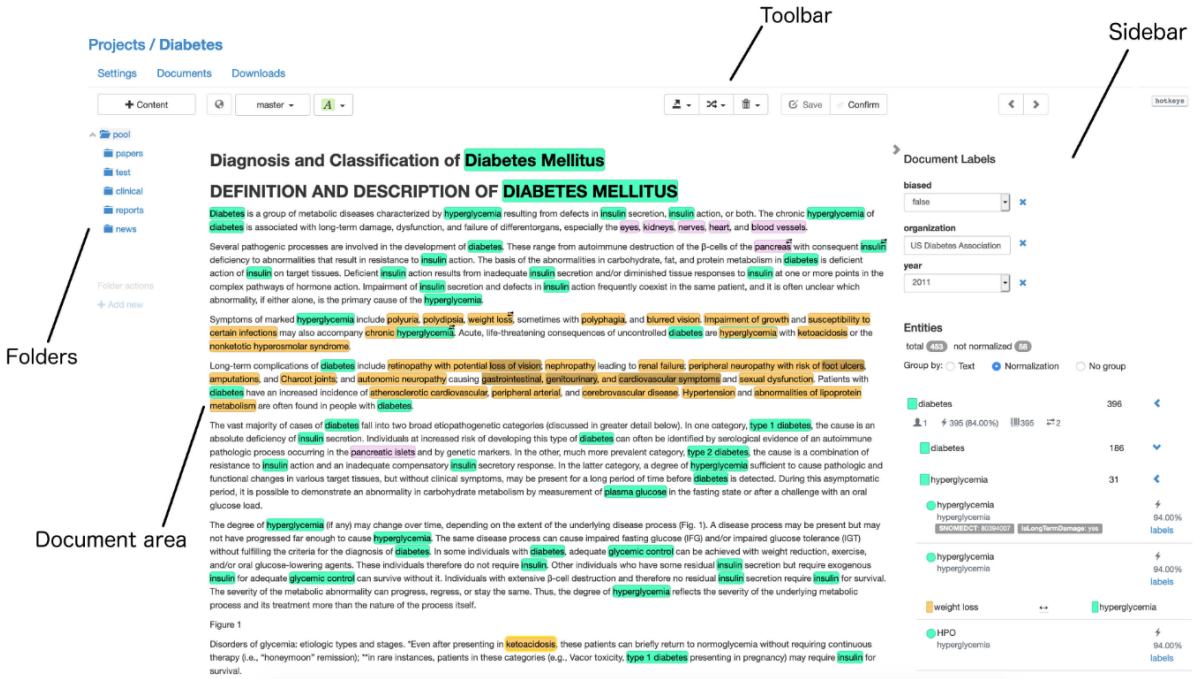


Figura 11: Exemplo do *TagTog*.

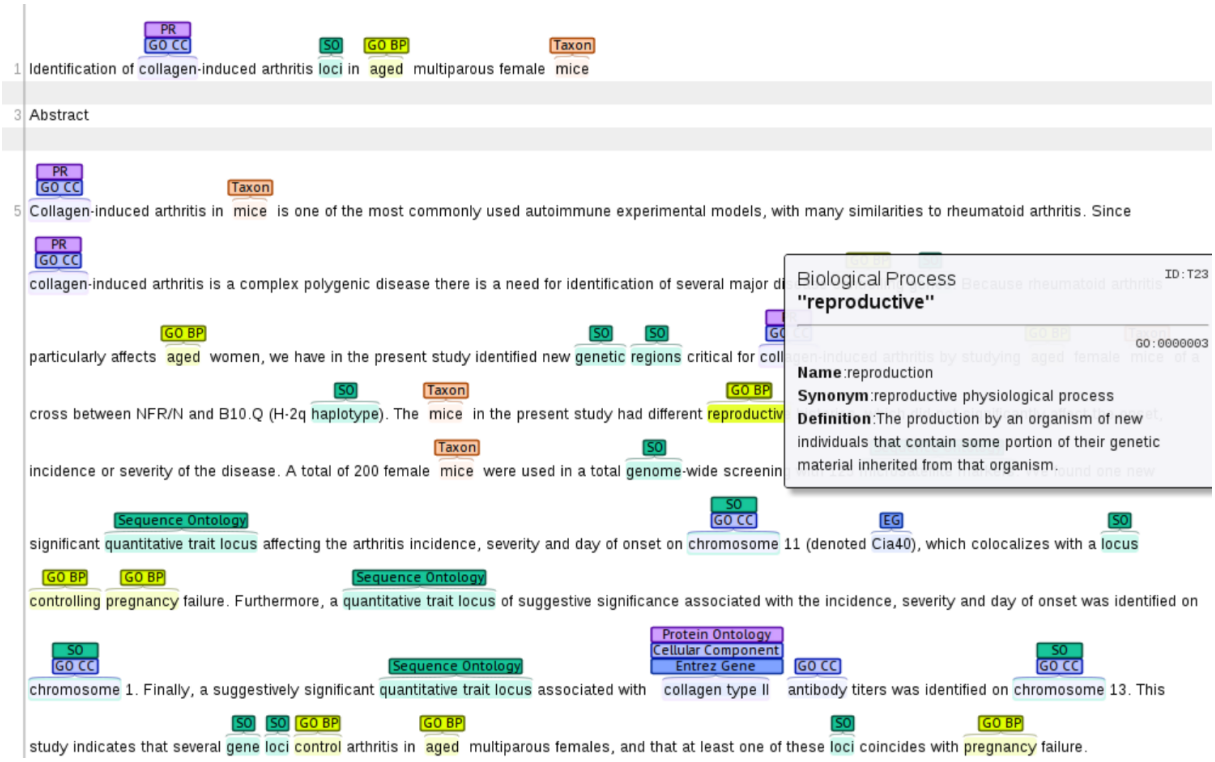


Figura 12: Exemplo do *Bart*.

Capítulo 3

Dados Clínicos, Modelos e Arquitetura

3.1 Base de Dados MIMIC-III

A *MIMIC-III* é uma enorme base de dados (Figura 13), de livre acesso, que contém dados não-identificados relacionados com a saúde de pacientes que foram admitidos em unidades de cuidados intensivos do *Beth Israel Deaconess Medical Center*. Esta versão da base de dados, a versão número três, contém especificamente dados de ocorrências compreendidas entre o ano 2001 e 2012. À data da escrita desta dissertação uma nova versão acabara de ser lançada, a *MIMIC-IV*, com dados compreendidos entre 2008 e 2019, mas que não foi possível utilizar pois as notas clínicas ainda não se encontravam acessíveis à comunidade e o suporte para esta era ainda bastante escasso [30] [31].

Apesar desta ser de livre acesso, e por se tratarem de dados sensíveis, todos os utilizadores que quiserem ter acesso à *MIMIC* têm de preencher um formulário de requisição. Além disso, o acesso só é garantido caso o utilizador: realize um curso certificado que preencha os requisitos pela *Health Insurance Portability and Accountability Act* (HIPAA) e assine um acordo de segurança, confiabilidade e utilização dos dados que vai receber.

De uma forma geral, o ambiente, os arquivos e a base de dados são vistos da seguinte forma:

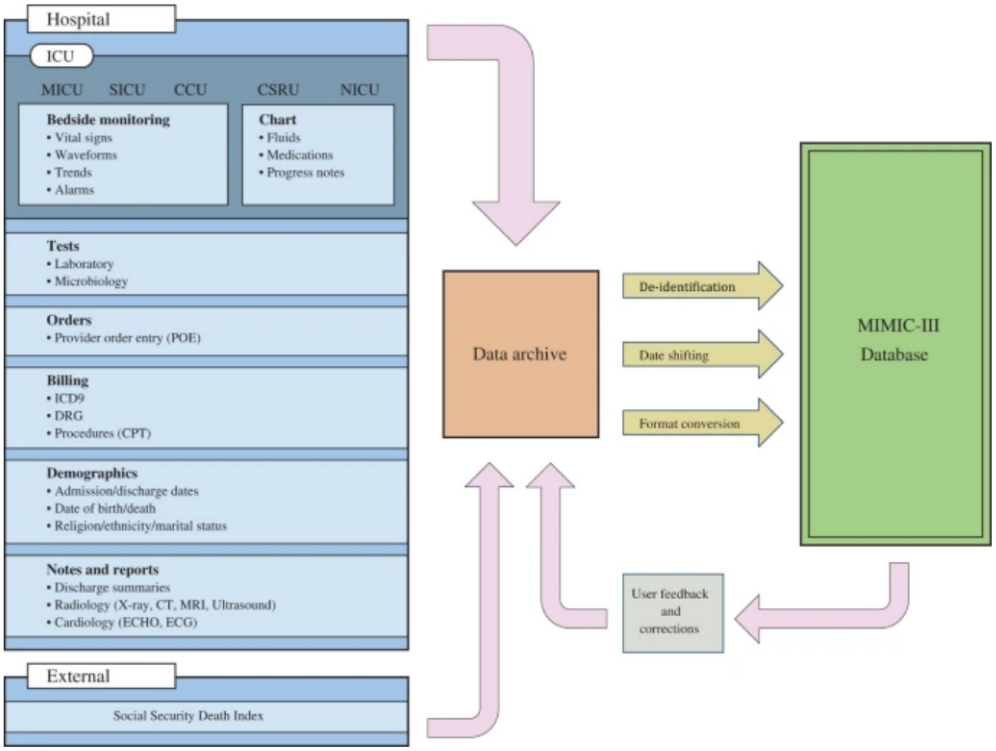


Figura 13: Vista geral da base de dados MIMIC-III.

Relativamente ao conteúdo em si, esta é então constituída por dados associados a mais de 53 mil admissões distintas. Todos os dados foram recolhidos durante o dia-a-dia, nunca tendo esta recolha algum tipo de interferência no trabalho dos profissionais de saúde ou vice-versa. Estes dados são relativos apenas e só a adultos com 16 anos ou mais que tenham dado entrada em unidades de cuidados intensivos. Outras estatísticas interessantes deste conjunto de dados são [31]:

- A mediana dos pacientes é de 65.8 anos;
- Dos pacientes admitidos, 55.9% são do sexo masculino;
- A mortalidade no hospital é de 11.5%.
- A mediana do tempo de hospitalização nas Unidade de Cuidados Intensivos (UCI) é de 2.1 dias, sendo que a mediana do tempo de hospitalização geral é de 6.9 dias.

Além de todos esses dados, a *MIMIC-III* contém ainda 4579 observações clínicas e 380 resultados laboratoriais [31].

Esta coleção de dados tem um impacto bastante positivo do ponto de vista da Informática por 2 grandes razões: primeiro porque os dados clínicos sempre foram de difícil acesso e utilização devido à sua sensibilidade e, em segundo, porque uma base de dados partilhada deste tipo permite a investigadores

e programadores percorrerem caminhos semelhantes através dos quais podem discutir, trocar ideias e resolver problemas em conjunto levando a que as pesquisas sejam cada vez mais produtivas e com melhores resultados.

A base de dados é composta por 43 tabelas que totalizam mais de 50 *gigabytes* de informação.

Table name	Description
ADMISSIONS	Every unique hospitalization for each patient in the database (defines HADM_ID).
CALLOUT	Information regarding when a patient was cleared for ICU discharge and when the patient was actually discharged.
CAREGIVERS	Every caregiver who has recorded data in the database (defines CGID).
CHARTEVENTS	All charted observations for patients.
CPTEVENTS	Procedures recorded as Current Procedural Terminology (CPT) codes.
D_CPT	High level dictionary of Current Procedural Terminology (CPT) codes.
D_ICD_DIAGNOSES	Dictionary of International Statistical Classification of Diseases and Related Health Problems (ICD-9) codes relating to diagnoses.
D_ICD_PROCEDURES	Dictionary of International Statistical Classification of Diseases and Related Health Problems (ICD-9) codes relating to procedures.
D_ITEMS	Dictionary of local codes ('ITEMIDs') appearing in the MIMIC database, except those that relate to laboratory tests.
D_LABITEMS	Dictionary of local codes ('ITEMIDs') appearing in the MIMIC database that relate to laboratory tests.
DATETIMEEVENTS	All recorded observations which are dates, for example time of dialysis or insertion of lines.
DIAGNOSES_ICD	Hospital assigned diagnoses, coded using the International Statistical Classification of Diseases and Related Health Problems (ICD) system.
DRGCODES	Diagnosis Related Groups (DRG), which are used by the hospital for billing purposes.
ICUSTAYS	Every unique ICU stay in the database (defines ICUSTAY_ID).
INPUTEVENTS_CV	Intake for patients monitored using the Philips CareVue system while in the ICU, e.g., intravenous medications, enteral feeding, etc.
INPUTEVENTS_MV	Intake for patients monitored using the iMDSoft MetaVision system while in the ICU, e.g., intravenous medications, enteral feeding, etc.
OUTPUTEVENTS	Output information for patients while in the ICU.
LABEVENTS	Laboratory measurements for patients both within the hospital and in outpatient clinics.
MICROBIOLOGYEVENTS	Microbiology culture results and antibiotic sensitivities from the hospital database.
NOTEEVENTS	Deidentified notes, including nursing and physician notes, ECG reports, radiology reports, and discharge summaries.
PATIENTS	Every unique patient in the database (defines SUBJECT_ID).
PRESCRIPTIONS	Medications ordered for a given patient.
PROCEDUREEVENTS_MV	Patient procedures for the subset of patients who were monitored in the ICU using the iMDSoft MetaVision system.
PROCEDURES_ICD	Patient procedures, coded using the International Statistical Classification of Diseases and Related Health Problems (ICD) system.
SERVICES	The clinical service under which a patient is registered.
TRANSFERS	Patient movement from bed to bed within the hospital, including ICU admission and discharge.

Figura 14: Visão geral da base de dados *MIMIC-III*.

A tabela que contém todas as ocorrências observadas em pacientes (*CHARTEVENTS*) está particionada na base de dados com o identificador respetivo a seguir ao nome.

Destas tabelas, a mais relevante para esta dissertação é a tabela *NOTEEVENTS*, que é a tabela das notas clínicas.

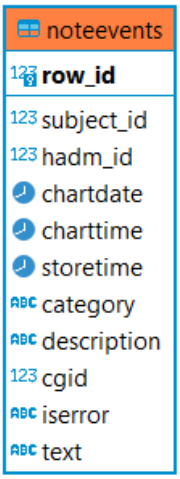
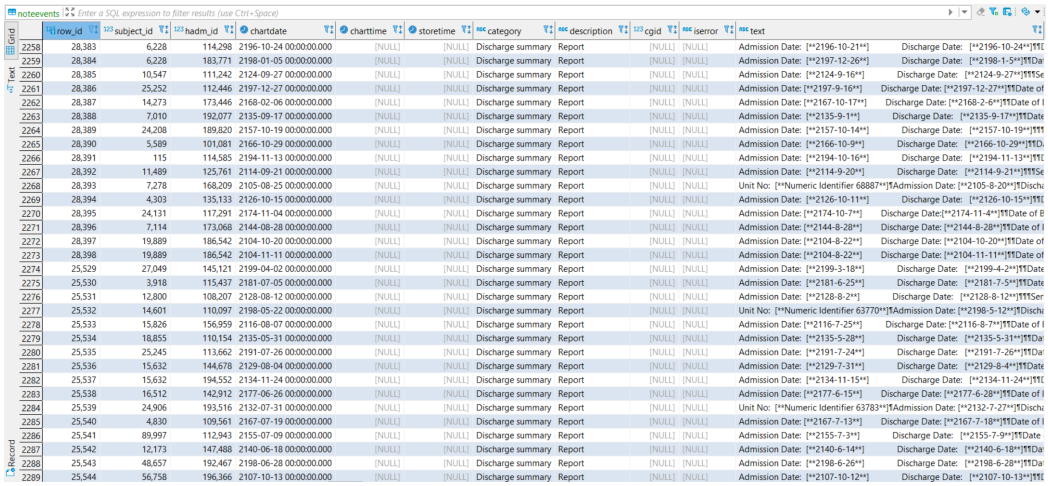


Figura 15: Esquema da tabela *NOTEEVENTS*.

- **ROW_ID:** identificador da linha da tabela;
- **SUBJECT_ID:** identificador do paciente, equivalente ao nº do processo em Portugal;
- **HADM_ID:** identificador da admissão, número do episódio;
- **CHARTDATE**: dia de admissão;
- **CHARTTIME:** hora de admissão;
- **STORETIME:** data de entrada da nota na base de dados;
- **CATEGORY:** categoria da nota clínica.
  - *Case Management*
  - *Consult*
  - *Discharge Summary*
  - *ECG*
  - *Echo*
  - *General*
  - *Nursing*
  - *Nursing/Other*
  - *Nutrition*
  - *Pharmacy*
  - *Physician*

  - *Radiology*
  - *Rehab Services*
  - *Respiratory*
  - *Social Work*

- **DESCRIPTION:** descrição da nota clínica, normalmente o tipo de exame efectuado ou algo que caracterize a nota;
- **CGID:** identificador de enfermeiro;
- **ISERROR:** binário que define se a nota clínica é erro ou não;
- **TEXT:** nota clínica.



	row_id	subject_id	hadm_id	chartdate	charttime	storetime	category	description	cgid	iserror	text
2258	28,383	6,228	114,298	2196-10-24 00:00:00.000	[NULL]	[NULL]	Discharge summary	Report	[NULL]	[NULL]	Admission Date: [**2196-10-21**] Discharge Date: [**2196-10-24**]¶¶
2259	28,384	6,228	183,771	2198-01-05 00:00:00.000	[NULL]	[NULL]	Discharge summary	Report	[NULL]	[NULL]	Admission Date: [**2197-12-26**] Discharge Date: [**2198-1-5**]¶¶Da
2260	28,385	10,547	111,242	2124-09-27 00:00:00.000	[NULL]	[NULL]	Discharge summary	Report	[NULL]	[NULL]	Admission Date: [**2124-9-16**] Discharge Date: [**2124-9-27**]¶¶¶Se
2261	28,386	25,252	112,446	2197-12-27 00:00:00.000	[NULL]	[NULL]	Discharge summary	Report	[NULL]	[NULL]	Admission Date: [**2197-9-16**] Discharge Date: [**2197-12-27**]¶¶Date of
2262	28,387	14,273	173,446	2168-02-06 00:00:00.000	[NULL]	[NULL]	Discharge summary	Report	[NULL]	[NULL]	Admission Date: [**2167-10-17**] Discharge Date: [**2168-2-6**]¶¶Date of I
2263	28,388	7,010	192,077	2135-09-17 00:00:00.000	[NULL]	[NULL]	Discharge summary	Report	[NULL]	[NULL]	Admission Date: [**2135-9-1**] Discharge Date: [**2135-9-17**]¶¶Date
2264	28,389	24,208	189,820	2157-10-19 00:00:00.000	[NULL]	[NULL]	Discharge summary	Report	[NULL]	[NULL]	Admission Date: [**2157-10-14**] Discharge Date: [**2157-10-19**]¶¶¶
2265	28,390	5,589	101,081	2166-10-29 00:00:00.000	[NULL]	[NULL]	Discharge summary	Report	[NULL]	[NULL]	Admission Date: [**2166-10-9**] Discharge Date: [**2166-10-29**]¶¶D
2266	28,391	115	114,585	2194-11-13 00:00:00.000	[NULL]	[NULL]	Discharge summary	Report	[NULL]	[NULL]	Admission Date: [**2194-10-16**] Discharge Date: [**2194-11-13**]¶¶
2267	28,392	11,489	125,761	2114-09-21 00:00:00.000	[NULL]	[NULL]	Discharge summary	Report	[NULL]	[NULL]	Admission Date: [**2114-9-20**] Discharge Date: [**2114-9-21**]¶¶Se
2268	28,393	7,278	168,209	2105-08-25 00:00:00.000	[NULL]	[NULL]	Discharge summary	Report	[NULL]	[NULL]	Unit No: [**Numeric Identifier 68887**]¶Admission Date: [**2105-8-20**]¶Dischа
2269	28,394	4,303	135,133	2126-10-15 00:00:00.000	[NULL]	[NULL]	Discharge summary	Report	[NULL]	[NULL]	Admission Date: [**2126-10-11**] Discharge Date: [**2126-10-15**]¶¶
2270	28,395	24,131	117,291	2174-11-04 00:00:00.000	[NULL]	[NULL]	Discharge summary	Report	[NULL]	[NULL]	Admission Date: [**2174-10-7**] Discharge Date:[**2174-11-4**]¶¶Date of E
2271	28,396	7,114	173,068	2144-08-28 00:00:00.000	[NULL]	[NULL]	Discharge summary	Report	[NULL]	[NULL]	Admission Date: [**2144-8-28**] Discharge Date: [**2144-8-28**]¶¶Date of I
2272	28,397	19,889	186,542	2104-10-20 00:00:00.000	[NULL]	[NULL]	Discharge summary	Report	[NULL]	[NULL]	Admission Date: [**2104-8-22**] Discharge Date: [**2104-10-20**]¶¶Date of
2273	28,398	19,889	186,542	2104-11-11 00:00:00.000	[NULL]	[NULL]	Discharge summary	Report	[NULL]	[NULL]	Admission Date: [**2104-8-22**] Discharge Date: [**2104-11-11**]¶¶Date of
2274	25,529	27,049	145,121	2199-04-02 00:00:00.000	[NULL]	[NULL]	Discharge summary	Report	[NULL]	[NULL]	Admission Date: [**2199-3-18**] Discharge Date: [**2199-4-2**]¶¶Date
2275	25,530	3,918	115,437	2181-07-05 00:00:00.000	[NULL]	[NULL]	Discharge summary	Report	[NULL]	[NULL]	Admission Date: [**2181-6-25**] Discharge Date: [**2181-7-5**]¶¶Date
2276	25,531	12,800	108,207	2128-08-12 00:00:00.000	[NULL]	[NULL]	Discharge summary	Report	[NULL]	[NULL]	Admission Date: [**2128-8-2**] Discharge Date: [**2128-8-12**]¶¶¶Ser
2277	25,532	14,601	110,097	2198-05-22 00:00:00.000	[NULL]	[NULL]	Discharge summary	Report	[NULL]	[NULL]	Unit No: [**Numeric Identifier 63770**]¶Admission Date: [**2198-5-12**]¶Dischа
2278	25,533	15,826	156,959	2116-08-07 00:00:00.000	[NULL]	[NULL]	Discharge summary	Report	[NULL]	[NULL]	Admission Date: [**2116-7-25**] Discharge Date: [**2116-8-7**]¶¶Date of I
2279	25,534	18,855	110,154	2135-05-31 00:00:00.000	[NULL]	[NULL]	Discharge summary	Report	[NULL]	[NULL]	Admission Date: [**2135-5-28**] Discharge Date: [**2135-5-31**]¶¶Da
2280	25,535	25,245	113,662	2191-07-26 00:00:00.000	[NULL]	[NULL]	Discharge summary	Report	[NULL]	[NULL]	Admission Date: [**2191-7-24**] Discharge Date: [**2191-7-26**]¶¶Da
2281	25,536	15,632	144,678	2129-08-04 00:00:00.000	[NULL]	[NULL]	Discharge summary	Report	[NULL]	[NULL]	Admission Date: [**2129-7-31**] Discharge Date: [**2129-8-4**]¶¶Date
2282	25,537	15,632	194,552	2134-11-24 00:00:00.000	[NULL]	[NULL]	Discharge summary	Report	[NULL]	[NULL]	Admission Date: [**2134-11-15**] Discharge Date: [**2134-11-24**]¶¶
2283	25,538	16,512	142,912	2177-06-26 00:00:00.000	[NULL]	[NULL]	Discharge summary	Report	[NULL]	[NULL]	Admission Date: [**2177-6-15**] Discharge Date: [**2177-6-28**]¶¶Date of I
2284	25,539	24,906	193,516	2132-07-31 00:00:00.000	[NULL]	[NULL]	Discharge summary	Report	[NULL]	[NULL]	Unit No: [**Numeric Identifier 63783**]¶Admission Date: [**2132-7-27**]¶Dischа
2285	25,540	4,830	109,561	2167-07-19 00:00:00.000	[NULL]	[NULL]	Discharge summary	Report	[NULL]	[NULL]	Admission Date: [**2167-7-13**] Discharge Date: [**2167-7-18**]¶¶Date of I
2286	25,541	89,997	112,943	2155-07-09 00:00:00.000	[NULL]	[NULL]	Discharge summary	Report	[NULL]	[NULL]	Admission Date: [**2155-7-3**] Discharge Date: [**2155-7-9**]¶¶Date
2287	25,542	12,173	147,488	2140-06-18 00:00:00.000	[NULL]	[NULL]	Discharge summary	Report	[NULL]	[NULL]	Admission Date: [**2140-6-14**] Discharge Date: [**2140-6-18**]¶¶Da
2288	25,543	48,657	192,467	2198-06-28 00:00:00.000	[NULL]	[NULL]	Discharge summary	Report	[NULL]	[NULL]	Admission Date: [**2198-6-26**] Discharge Date: [**2198-6-28**]¶¶Da
2289	25,544	56,758	196,366	2107-10-13 00:00:00.000	[NULL]	[NULL]	Discharge summary	Report	[NULL]	[NULL]	Admission Date: [**2107-10-12**] Discharge Date: [**2107-10-13**]¶¶

Figura 16: Vista da tabela *NOTEEVENTS* no *DBeaver*.

Desta tabela, a coluna que vai ser mais utilizada nesta dissertação é, sem dúvida, a que contém o texto clínico sobre o paciente, no qual serão aplicados modelos e será feito o tratamento do texto de forma a auxiliar os profissionais de saúde.

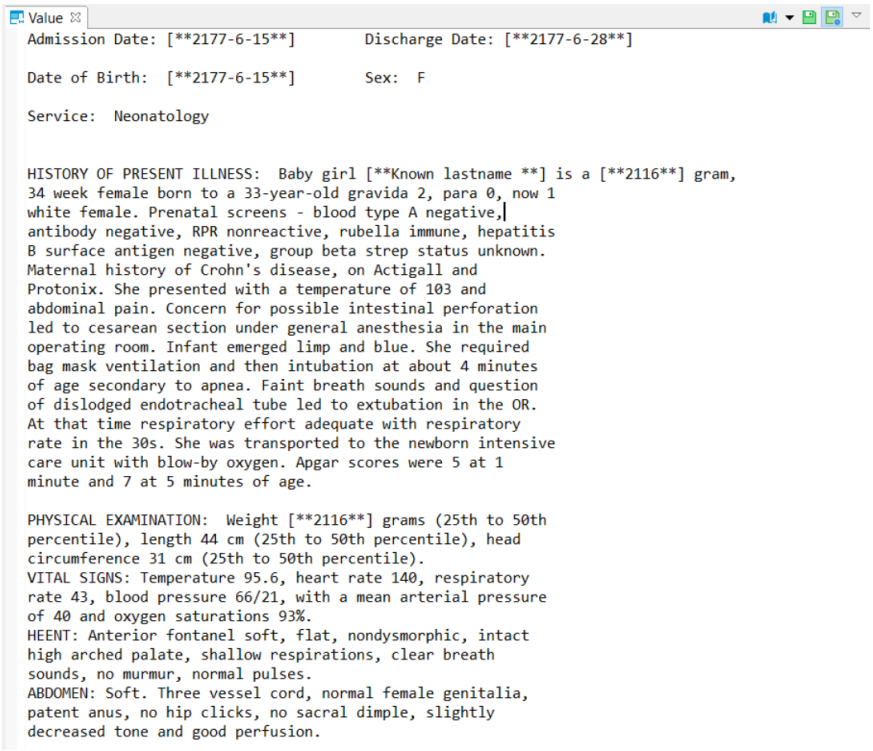


Figura 17: Exemplo de uma coluna *TEXT* da tabela *NOTEEVENTS*.

3.2 Técnicas e Modelos Pré-Treinados

O PLN é uma área bastante diversificada, com inúmeras tarefas distintas que necessitam de grandes quantidades de dados anotados pelo humano para que os modelos possam ser treinados. Porém, muitas das vezes, apenas estão disponíveis algumas dezenas ou poucas centenas desses exemplos, tornado mais difícil o treino destes.

Para encurtar este distanciamento na área e aproximar os modelos dos dados de treino, os investigadores têm desenvolvido várias técnicas de treino geral utilizando apenas enormes quantidades de texto livre disponível na *web*. Com a resolução dessa lacuna de dados, é possível, com algumas mudanças nos modelos, aplicar os modelos em *datasets* mais pequenos e aumentar a precisão em tarefas como análise de sentimentos comparativamente a um modelo treinado a partir de um *dataset* construído de raiz [32].

Nos anos anteriores a 2017, com a introdução dos *Transformers* pela Google, os modelos utilizavam essencialmente *Recurrent Neural Networks* (RNN) e *Conventional Neural Networks* (CNN) para resolver tarefas de PLN. O *Transformer* desbloqueou um novo caminho para a construção de modelos e para a resolução de tarefas. Este não necessita que sequências de dados sejam processadas numa ordem fixa ao contrário dos modelos mais utilizados na altura. O facto deste novo elemento permitir que o processamento de texto fosse feito "de qualquer maneira" permitiu que os modelos que o utilizassem pudessem ser treinados em enormes quantidades de dados que até então não tinha sido possível [33].

Em 2018, a *Google* lançou o modelo pré-treinado *BERT* que revolucionou o mundo geral do PLN. O *BERT* é uma *framework open-source* desenvolvida para que os modelos computacionais compreendessem melhor o significado de uma linguagem num texto, utilizando como base para essa resolução o mesmo texto que envolve a palavra, de forma a estabelecer um contexto. Este modelo foi pré-treinado utilizando apenas texto livre do Wikipédia. Uma das grandes vantagens deste modelo é que permite ser ajustado para tarefas mais específicas utilizando apenas *datasets* mais pequenos de pergunta e resposta [34] [35].

O facto deste modelo ser bidirecional foi também uma grande inovação nesta área do ML. Na era pré-BERT, os modelos linguísticos apenas conseguiam ler sequencialmente. Com esta constante evolução dos *Transformers*, os cientistas da *Google* possibilitaram um grande aumento da capacidade de reconhecimento de contexto e resolução de ambiguidade numa linguagem [36] [35] [37].

Este modelo, atualmente, é utilizado em inúmeras tarefas do PLN, como:

- *Question-answering*;
- Sumarização;
- Previsão de frases;
- Polissemia e correferência (resolver problemas de palavras que pareçam ou soem de forma idêntica mas tenham diferentes significados);
- Resolução da ambiguidade;
- Classificação de sentimentos.

O *BERT* é um modelo *open-source* pelo que qualquer pessoa pode aceder e utilizar o mesmo. Segundo a *Google*, os utilizadores podem, em 30 minutos, treinar um sistema de pergunta e resposta com apenas um *Tensor Processor Unit* (TPU) e, em apenas algumas horas, com a utilização de um *Graphic Processor Unit* (GPU). Devido a isto, o avanço nesta área da Informática aumentou exponencialmente e permitiu que várias organizações, grupos de pesquisa ou até cientistas por conta própria desenvolvem-se vários modelos baseados em BERT especializados em certas tarefas, simplesmente através da aplicação de treino específico, em certos ambientes contextuais [34]. Alguns exemplos destes modelos são:

- **patentBERT**: classificação de patentes [38];
- **docBERT**: classificação de documentos [39];
- **bioBERT**: modelo de representação pré-treinado para mineração de texto biomédico [40];
- **sciBERT**: classificação de texto científico [41].

Resumindo o impacto do nascimento do *BERT*, a resolução de um qualquer problema de PLN passou agora para um processo de 2 tarefas simples [35] [37]:

1. Treinar um modelo numa enorme quantidade de texto livre (que pode ser supervisionado ou não), modelo este que pode ser transferido.

2. Afinar este modelo para tarefas específicas utilizando o conhecimento grande que já tem (supervisionado)

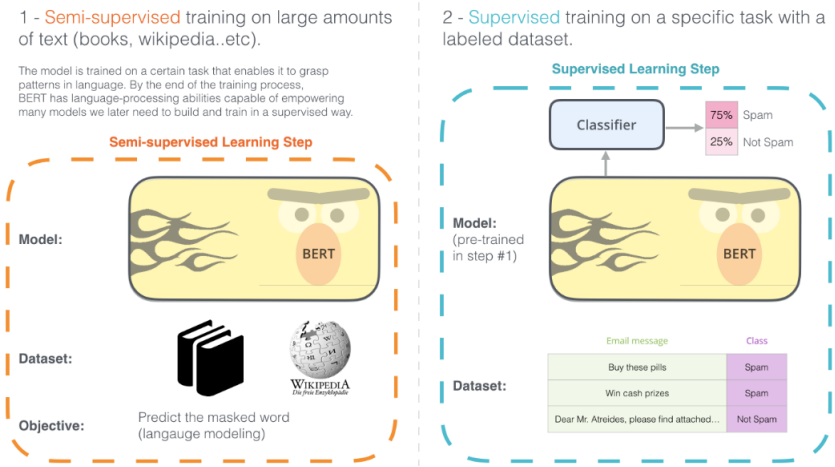


Figura 18: Desenvolvimento de um modelo *BERT*.

Sendo o tema desta dissertação algo já bastante debatido na comunidade de PLN, e de acordo com as pesquisas que foram levadas a cabo, chegou-se à conclusão que desenvolver um modelo destes a partir do zero seria, no fundo, desenvolver algo já existente [42].

Um modelo pré-treinado é um modelo que já foi treinado para uma tarefa específica com um *dataset* específico [43]. O uso deste tipo de modelos apresenta vantagens como:

- Em termos computacionais, criar e treinar um modelo de raiz para *MIMIC-III*, nomeadamente, para a tabela *NOTEEVENTS* que possui mais de 2 milhões de linhas seria bastante exigente e não-alcançável pela maioria das máquinas e num modelo pré-treinado com a utilização de grandes máquinas, isso já foi realizado;

- A quantidade de tempo que iria ser dispensado a treinar um modelo de raiz com os inúmeros cálculos e tentativas que seriam necessárias para apresentar resultados satisfatórios não seria compensatória proporcionalmente aos resultados que se obteriam comparativamente aos modelos pré-treinados [42].

Como o *dataset* a utilizar seria a *MIMIC-III* a quantidade de modelos pré-treinados aumentou, pois esta base de dados é amplamente utilizada pela comunidade de PLN. Sendo este *dataset* bastante popular no

meio, o passo seguinte foi analisar os modelos pré-treinados já existentes e decidir quais seriam os mais apropriados e teriam mais impacto na aplicação.

De acordo com uma sondagem do *Gradient Flow* as 2 bibliotecas de PLN mais utilizadas pelas organizações de saúde, em 2020, eram: *SparkNLP* e *spaCy* [44] [45].

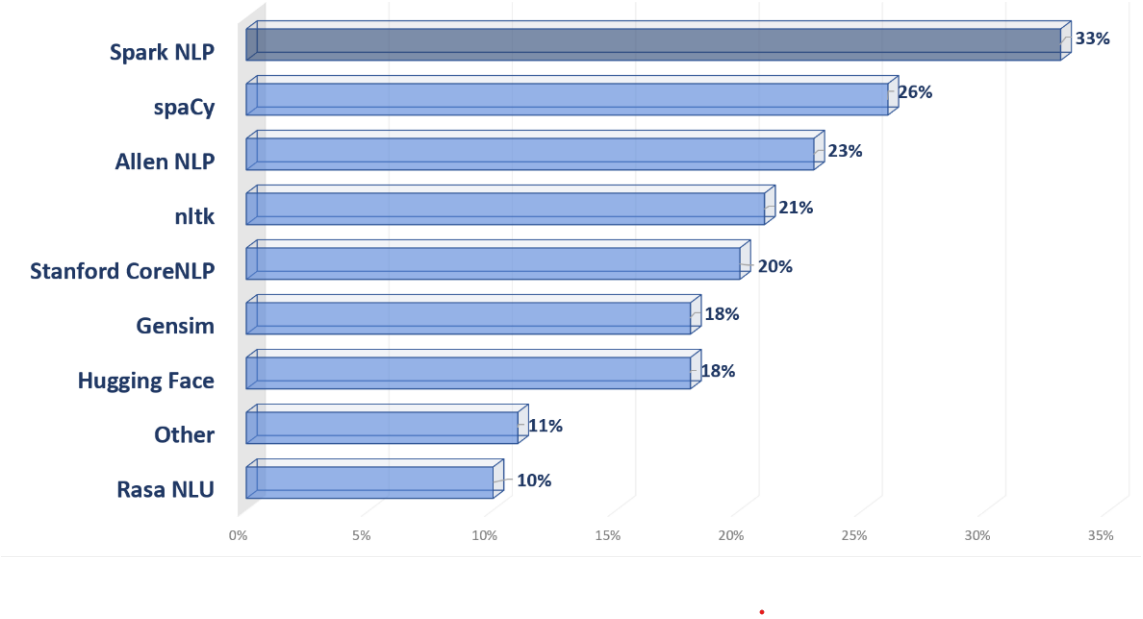


Figura 19: Utilização de bibliotecas de PLN em organizações de Saúde [46].

Para escolher entre as 2 grandes bibliotecas mais utilizadas pelas organizações a pesquisa passou por avaliar as 2 em relação a alguns parâmetros, tais como: precisão, performance, quantidade de modelos e suporte.

Em termos de precisão, segundo um estudo feito pela *Analytics India Magazine* em 2019, o *Spark NLP* apresentava metade dos erros que o *spaCy* em modelos de *Named Entity Recognition* (Figura 20).

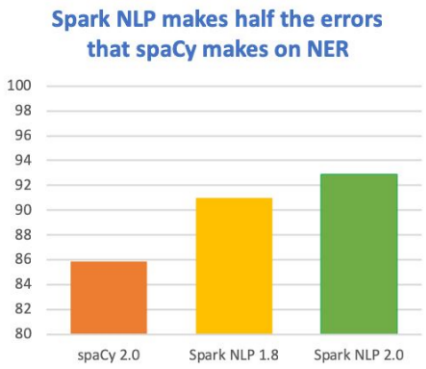


Figura 20: Comparação de erros em NER [47].

Noutro artigo, escrito em 2020, cujo objetivo era realizar uma comperação realista entre as 2 bibliotecas líderes, todos os modelos *Spark NLP* apresentaram maior precisão que os do *spaCy* (Figura

21).

SYSTEM	YEAR	LANGUAGE	ACCURACY
Spark NLP v2.4	2020	Python/Scala/Java/R	**93.3 (test F1) - 95.9 (dev F1)**
Spark NLP v2.x	2019	Python/Scala/Java/R	**93**
Spark NLP v1.x	2018	Python/Scala/Java/R	92
spaCy v2.x	2017	Python/Cython	92.6
spaCy v1.x	2015	Python/Cython	91.8
ClearNLP	2015	Java	91.7
CoreNLP	2015	Java	89.6
MATE	2015	Java	92.5
Turbo	2015	C++	92.4

Figura 21: *Benchmarking* dos 2 modelos [48].

No mesmo artigo, foram comparadas as velocidades de processamento das 2 bibliotecas e os resultados foram os seguintes (Figura 22):

	Start Memory (GB)	End Memory (GB)	Peak Memory (GB)	Elapsed Time (sec)
spaCy	2.7	5.4	7.1	1060-1080
Spark NLP	2.6	4.6	6.2	580

Figura 22: Comparação das velocidades de processamento das bibliotecas *Spark NLP* e *spaCy*.

Concluindo, os modelos do *Spark NLP* utilizam menos memória, são 2 vezes mais rápidos e, juntando a estes parâmetros de performance, estes ainda possuem mais precisão.

Em termos de quantidade de modelos disponíveis, no mesmo artigo referenciado em cima, da revista indiana de PLN, a oferta da biblioteca desenvolvida pelo *John Snow Labs* era bastante mais completa e mais avançada em relação às bibliotecas concorrentes. (Figura 23)

Name	NLTK	spaCy	CoreNLP	Spark NLP
Sentence Detection	Yes	Yes	Yes	Yes
Tokenization	Yes	Yes	Yes	Yes
Stemming	Yes	Yes	Yes	Yes
Lemmatization	Yes	Yes	Yes	Yes
POS Tagger	Yes	Yes	Yes	Yes
NER	Yes	Yes	Yes	Yes
Dependency Parse	Yes	Yes	Yes	Yes
Text Matcher	No	No	Yes	Yes
Date Matcher	No	No	Yes	Yes
Chunking	Yes	Yes	Yes	Yes
Spell Checker	No	No	No	Yes
Sentiment Detector	No	No	Yes	Yes
Pre-trained Models	Yes	Yes	Yes	Yes
Training Models	Yes	Yes	Yes	Yes

Figura 23: Modelos oferecidos pelas diferentes organizações em 2019 [47].

Além de todas estas comparações e para perceber realmente o impacto desta biblioteca na área, a biblioteca *Spark NLP* contabiliza cerca de 10 milhões de transferências totais desde o seu primeiro lançamento e mais de 1 milhão destas feitas só no mês de Outubro do ano 2021 (Figura 24). É importante salientar que estes números referem-se apenas às bibliotecas da linguagem *Python*.

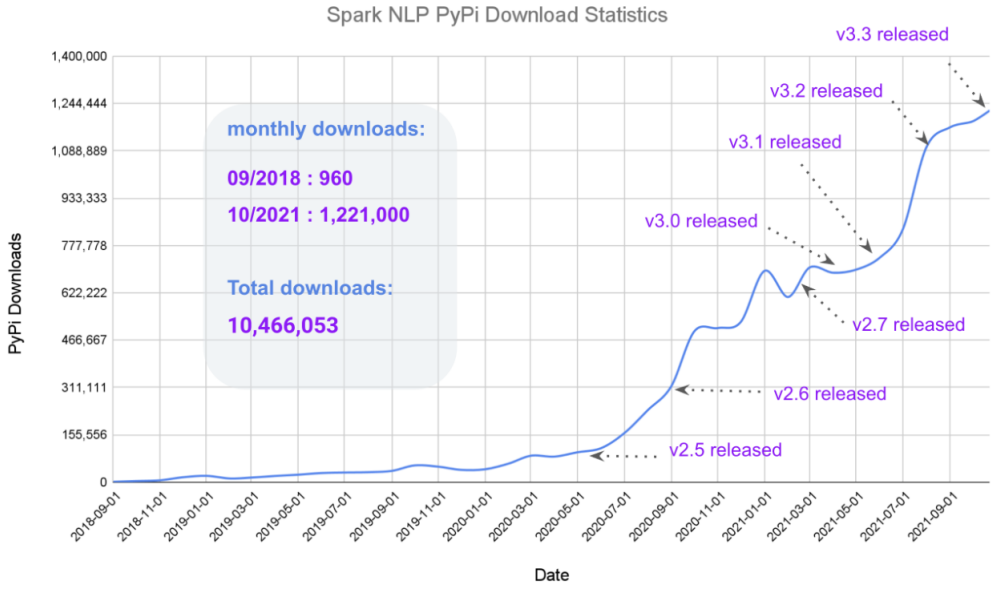


Figura 24: Gráfico de transferências das bibliotecas *Spark NLP*.

3.3 Recolha de Requisitos

Esta secção tem como objetivo especificar todos os requisitos da aplicação desenvolvida sejam eles funcionais ou não-funcionais cuja presença permite a construção de uma aplicação com vista a ajudar os profissionais de saúde a desempenhar as suas funções na avaliação e diagnóstico dos seus pacientes.

3.3.1 Requisitos Funcionais

Os requisitos funcionais enumeram as funcionalidades que se esperam que o sistema tenha. A aplicação deve:

- Permitir o *upload* de uma nota clínica em formato ficheiro ou em texto livre;
- Permitir a aplicação de modelos clínicos a notas clínicas carregadas pelo utilizador;
- Permitir a visualização dos resultados com *highlight* de texto relevante para profissionais de saúde consoante a presença ou ausência de certas características;
- Permitir a conexão a uma base de dados local;
- Permitir acesso a uma lista de notas clínicas de um paciente especificado pelo profissional de saúde;
- Permitir acesso a uma nota clínica da base de dados local;
- Permitir aplicação de modelos clínicos a uma nota clínica selecionada da base de dados local pelo profissional de saúde;
- Permitir a autenticação de um médico na aplicação (*login*, registo e *logout*);
- Não permitir carregamento de ficheiros que não *.txt*;
- Não permitir inserção de notas clínicas carregadas pelo utilizador na base de dados local.

3.3.2 Requisitos Não Funcionais

Para garantir uma aplicação de qualidade, são definidos estes requisitos que estão relacionados com o uso da aplicação de acordo com certos parâmetros como: desempenho, usabilidade, segurança, disponibilidade, manutenção e tecnologias envolvidas. Assim, os requisitos não funcionais são:

- Disponibilização de uma aplicação *web*;

- Disponibilização de modelos clínicos da biblioteca *open-source* de processamento de linguagem natural para *Python*, *Spark NLP*;
- A aplicação *web* deve ter uma aparência simplista e agradável;
- A aplicação *web* deve possuir fácil usabilidade;
- O *upload* de uma nota clínica em formato livre deve ser fácil;
- O acesso às notas clínicas da base de dados deve ser fácil.

3.4 Arquitectura

3.4.1 *Backend*

Para o desenvolvimento desta aplicação, a opção escolhida para o *backend* foi o *Flask*. Foram consideradas outras *frameworks* como o *Django* e o *Node.js*, com as quais também seria possível realizar este projeto. Contudo, estas não possuem uma curva de aprendizagem tão simples e uma implementação fácil como o *Flask* [49].

Apesar de já ter trabalho com outras *frameworks*, que também poderiam ser utilizadas na realização desta aplicação, como o caso do *Django* ou o *Node.js*, que foram também consideradas e avaliadas, decidi optar por experimentar algo com o qual nunca tivesse trabalhado e que, ao mesmo tempo aliado à curiosidade de ser algo novo, proporcionasse todas as ferramentas necessárias ao desenvolvimento desta *app*, com uma curva de aprendizagem simples e de fácil implementação [49].

O *Flask* é uma *framework* leve que permite aos utilizadores terem um controlo total no desenvolvimento da aplicação. Possui uma usabilidade e aprendizagem simples com uma curva de aprendizagem bastante diferente do maior concorrente. O facto de não possuir nenhuma funcionalidade de administração (como o *Django*) não foi um problema porque seria algo que não iria ser necessário à partida [50] [51].

Como uma das etapas principais desta aplicação seria trabalhar com uma base de dados enorme, a liberdade que o *Flask* oferece neste tema com a utilização de bibliotecas, como o *SQLAlchemy*, foi também um ponto a favor [52].

3.4.2 *Frontend*

As tecnologias apresentadas nesta subsecção e foram utilizadas nesta aplicação são responsáveis pelo desenvolvimento da parte da aplicação que está em contacto direto com o utilizador. No fundo, a parte gráfica deste projeto é o resultado da aplicação das tecnologias descritas nesta secção.

Nesta aplicação, as 2 tecnologias utilizadas para esta componente foram:

- **HTML:** responsável pela estrutura de cada página da aplicação que o utilizador visualiza.
- **CSS:** responsável por ”dar vida”à estrutura da página, com a aplicação de estilos.

Além destas maiores componentes, foi também utilizado o mecanismo de *templates* para *webpages*, *Jinja2*. Este mecanismo de template para a linguagem de programação *Python*, lançado em 2008, permite uma maior personalização das páginas *HTML* e é o mecanismo padrão do *Flask* [53]. Esta ferramenta apresenta uma grande flexibilidade e permitiu, nesta aplicação, uma melhor comunicação entre o *backend* e o *frontend* em tarefas como renderização de tabelas ou notas clínicas.

3.5 Escolha de Modelos

A escolha dos modelos para a aplicação era uma etapa importante. Tendo em conta que um dos objetivos da dissertação era facilitar e reduzir a carga de trabalho dos médicos em tarefas superficiais ou análises de informações redundantes foi então necessário perceber quais os modelos que poderiam ter mais impacto nestas tarefas.

Para esta aplicação foram escolhidos 3 tipos de modelos: *Spell Checker*, *Clinical Assertion* e *Clinical Entity Resolution*. É de salientar que devido à arquitetura da aplicação *web* e ao modo como esta foi desenvolvida a adição de outros tipos de modelos da biblioteca *Spark NLP* é relativamente fácil e algo que poderá ser feito no futuro para tornar a aplicação mais completa e útil aos profissionais de saúde.

Tendo em conta que muitas vezes devido à sobrecarga de trabalho, ou simplesmente devido a erros de escrita rápida no teclado, muitas notas médicas podem ser inseridas com erros ortográficos, o primeiro modelo que escolhi visa corrigir essas notas de forma a poderem ser melhor trabalhadas e analisadas posteriormente. Este modelo foi treinado em notas clínicas da *Informatics for Integrating Biology & the Bedside (i2b2)* e texto da *PubMed*.

A principal tarefa de um médico quando analisa relatórios e notas clínicas de um paciente com um histórico passa por perceber que tipo de tratamentos este já efetuou e quais foram os resultados, o histórico familiar e pessoal e também, claro, as doenças que apresenta o seu histórico. Como tal, o modelo que fez mais sentido aplicar e que é, provavelmente, aquele cujo resultado é de maior utilidade a um profissional de saúde foi o modelo de *Clinical Assertion* que dado uma nota clínica deteta a presença ou a ausência de termos médicos sejam eles doenças, tratamentos ou exames. Este modelo tem como *output* uma tabela composta por 3 colunas: *chunks*, ou seja, os termos médicos detetados; *type*, que define o tipo de termo médico e *assertion* que contém o resultado (presente, ausente ou hipotético) desse termo médico.

Como descrito inicialmente nesta dissertação, a codificação ICD-10 está bastante presente no panorama da Medicina em Portugal. É esta que permite o aumento da riqueza e da qualidade dos dados

clínicos. A utilização desta terminologia, muito próxima da linguagem clínica, que é uma linguagem caracterizada pela apresentação de características únicas como frases sem verbos, pouca pontuação e bastantes abreviações, em mais de 27 países proporciona uma grande qualidade de comunicação entre todo o tipo de instituição de saúde quer nacionais, quer internacionais. Por este motivo, fazia sentido a aplicação possuir algo que poupasse trabalho ao profissional de saúde e permitisse aumentar, de forma mais fácil e eficiente, a comunicação entre instituições clínicas com a utilização desta terminologia. Para isso, surgiu então um modelo *Clinical Entity Resolution*. Este modelo, dado uma nota clínica, retorna, para todos os termos médicos detetados, a respetiva codificação ICD-10 e a possível resolução da mesma. Assim, além do médico poder perceber facilmente o que o paciente já teve/fez ou não, pode facilmente ter acesso ao código dessa entidade para poder trabalhar e comunicar.

Dentro destas 3 categorias de modelos, foi também necessário escolher a versão do modelo a utilizar. As bibliotecas *Spark NLP* apresentam várias opções dentro da mesma qualidade consoante o tipo de treino que tiveram. Por exemplo, na gama dos *Clinical Entity Resolvers* existem 15 versões diferentes que utilizam a variante *SBERT* (Figura 25). No total existem cerca de 33 modelos relacionados com a resolução de códigos ICD-10. No total, contabilizando todas as nomenclaturas médicas existentes, estas bibliotecas oferecem 103 modelos e *pipelines*

```
documentAssembler = DocumentAssembler()\
  .setInputCol("text")\
  .setOutputCol("ner_chunk")

sbert_embedder = BertSentenceEmbeddings\
      .pretrained('sbiobert_base_cased_mli', 'en','clinical/models')\
      .setInputCols(["ner_chunk"])\
      .setOutputCol("sbert_embeddings")

snomed_ct_resolver = SentenceEntityResolverModel\
  .pretrained("sbiobertresolve_snomed_findings","en", "clinical/models") \
  .setInputCols(["ner_chunk", "sbert_embeddings"]) \
  .setOutputCol("snomed_code")\
  .setDistanceFunction("EUCLIDEAN")

snomed_pipelineModel = PipelineModel(
    stages = [
        documentAssembler,
        sbert_embedder,
        snomed_ct_resolver])

snomed_lp = LightPipeline(snomed_pipelineModel)
```

- sbiobertresolve_icd10cm
- sbiobertresolve_icd10cm_augmented
- sbiobertresolve_icd10pcs
- sbiobertresolve_snomed_findings (with clinical_findings concepts from CT version)
- sbiobertresolve_snomed_findings_int (with clinical_findings concepts from INT version)
- sbiobertresolve_snomed_auxConcepts (with Morph Abnormality, Procedure, Substance, Physical Object, Body Structure concepts from CT version)
- sbiobertresolve_snomed_auxConcepts_int (with Morph Abnormality, Procedure, Substance, Physical Object, Body Structure concepts from INT version)
- sbiobertresolve_rxnorm
- sbiobertresolve_rxcui
- sbiobertresolve_icdo
- sbiobertresolve_cpt
- sbiobertresolve_cpt_augmented
- sbiobertresolve_cpt_procedures_augmented
- sbiobertresolve_hcc_augmented
- sbiobertresolve_icd10cm_hcc_augmented

Figura 25: Modelos existentes na biblioteca *SparkNLP* que utilizam BERT.

De acordo com o delineado em secções iniciais desta dissertação, a opção passou então por um modelo de resolução ICD-10 e, como enunciado na secção 3.2, que aplicasse conceitos *BERT* no seu treino. O modelo escolhido foi o *sbiobertresolve_icd10cm_augmented* que é uma versão aumentada da versão normal deste modelo que apesar de ocupar mais espaço e ser mais robusta compensa com uma resolução bastante mais completa.

A nível de correção gramatical, a oferta é simples mas eficaz, existem apenas 8 modelos diferentes (para a língua inglesa) cujas diferenças entre eles se baseia no tipo de texto médico que queremos corrigir.

Capítulo 4

Implementação

O principal capítulo desta dissertação. Neste, é descrito todo o processo que levou ao resultado final.

4.1 Organização da Aplicação

De forma geral a estrutura da aplicação divide-se então pelo *backend* e pelo *frontend* (Figura 26). A pasta *templates* contém todos os ficheiros *.html* relativamente à estrutura das *web pages* e a pasta *static* contém todos os estilos utilizados pelos ficheiros estruturais. Na parte do *backend* a aplicação divide-se da seguinte maneira:

- *app.py:* este ficheiro inicia a aplicação. É neste ficheiro que é lançada a base de dados que contém os utilizadores assim como as bibliotecas necessárias à gestão de *logins*.
- *main.py:* o ficheiro principal da aplicação. Neste, é feito todo o trabalho relacionado com a biblioteca *Spark NLP* desde o início de sessão até à aplicação dos modelos. Todas as rotas relacionadas com o projeto estão também aqui presentes.
- *models.py:* aqui está contido o modelo *User*.

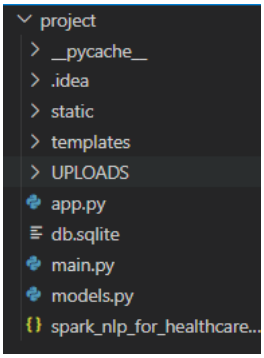


Figura 26: Estrutura da Aplicação.

Dentro da pasta do projeto encontra-se também a pasta *UPLOADS* que contém todas as notas clínicas carregadas pelos utilizadores.

4.1.1 Bases de Dados e Ligações

Com a utilização de 2 bases de dados pensou-se que seria necessária a utilização do *binding* da biblioteca *SQLAlchemy*. Esta funcionalidade permite a utilização de 2 ou mais conexões em simultâneo a diferentes bases de dados, consoante especificação no código. Porém, após análise percebeu-se que não seria necessário pois a conexão à base de dados *MIMIC* seria pontual e apenas seria necessária uma conexão permanente à base de dados dos utilizadores para efeitos de gestão do *login*.

A opção passou então pela utilização de uma base de dados simples do tipo *SLQLite* para os utilizadores. Quanto à base de dados *MIMIC*, não houve a possibilidade de tomar uma opção pois os *scripts* disponibilizados pela *PhysioNet*, empresa responsável pelo *dataset*, estavam orientados para o *PostgresSQL*.

Para a base de dados local de utilizadores apenas foi necessária a criação de uma tabela.

Lista de Tabelas 4.1: Modelo *User* da aplicação

```
1 class User(UserMixin, db.Model):
2     id = db.Column(db.Integer, primary_key=True) # primary keys are required by SQLAlchemy
3     medical_id = db.Column(db.Integer, unique=True)
4     email = db.Column(db.String(100), unique=True)
5     password = db.Column(db.String(100))
6     name = db.Column(db.String(1000))
```

4.1.2 *Upload* de notas clínicas

O *upload* de notas clínicas na aplicação é feito de forma bastante simples e intuitiva para qualquer utilizador desta e pode ser feito de 2 maneiras diferentes. A primeira, carregando um ficheiro *.txt* e, a segunda, escrevendo a própria nota clínica.

No primeiro caso, é guardado o nome do ficheiro para permitir, aquando da aplicação de um modelo, localizar o texto onde será feita essa aplicação. Todos os ficheiros carregados são guardados no mesmo *path*.

Lista de Tabelas 4.2: Carregamento de uma nota clínica do tipo ficheiro para a aplicação

```
@app.route('/upload', methods=['POST'])
@login_required
def upload_file():
   if request.method == 'POST':
      if 'file' not in request.files:
         flash('No file part')
         return redirect(request.url)
      file = request.files['file']
      if file.filename == '':
         flash('No file selected for uploading')
         return redirect(request.url)
      if file and allowed_file(file.filename):
         filename = secure_filename(file.filename)
         file.save(os.path.join(app.config['UPLOAD_FOLDER'], filename))
         nota = open_show("C:/Users/ricar/Desktop/TESE/automatic-annotation-tool/UPLOADS/" +
             ↪ file.filename)
         return render_template('filenote.html', content=nota, filename=file.filename)
      else:
         flash('Allowed file types are txt')
         return redirect(request.url)
```

Quanto ao carregamento por escrito, este é guardado num ficheiro, dentro da mesma pasta que os ficheiros carregados pelo mesmo motivo anterior.

Lista de Tabelas 4.3: Carregamento de uma nota clínica por escrito para a aplicação

```
@app.route('/upload_self', methods=['POST'])
@login_required
def self_note():
   if request.method == 'POST':
      text = request.form['text']
      fo = open("C:/Users/ricar/Desktop/TESE/automatic-annotation-tool/UPLOADS/selfnote.txt","
          ↪ w+")
      fo.write(text)
      fo.close()
      nota = get_content("C:/Users/ricar/Desktop/TESE/automatic-annotation-tool/UPLOADS/
          ↪ selfnote.txt")
      return render_template('selfnote.html', content=nota)
```

4.1.3 Modelos *SparkNLP*

4.1.3.1 Carregamento dos Modelos

O *Spark NLP* é uma biblioteca *open-source*. Como tal, qualquer utilizador pode aceder à mesma. A extensão comercial *Spark NLP for Healthcare* estende esta biblioteca ao uso em texto clínico e biomédico. É nesta biblioteca que se encontram os modelos pré-treinados que foram aplicados neste projeto.

Para aplicar estes modelos, a primeira etapa passa então pelo carregamento dos modelos da biblioteca. Estes modelos são compostos por outros modelos, ou seja, um modelo de *Spell Checking*, por exemplo, além de possuir o modelo principal responsável pela tarefa enunciada, tem na sua *pipeline* outros modelos auxiliares ao processo como:

- **DocumentAssembler**
- **Tokenizer**
- **SpellModel**
- **Finisher**

Assim, o carregamento é feito por esta ordem o que resulta numa *NlpPipeline*, que, como o nome indica, vai ser a responsável pelo processamento de dados, originando depois o *output* que é previamente especificado no carregamento do modelo principal, ou seja, neste caso seria especificado quando o modelo *SpellModel* é carregado.

Lista de Tabelas 4.4: Exemplo do carregamento do modelo de *Clinical Assertion*

```
1  def load_clinical_assertion():
2
3     documentAssembler = DocumentAssembler() .setInputCol("text") .setOutputCol("document")
4     sentenceDetector = SentenceDetector() .setInputCols(["document"]) .setOutputCol("sentence")
5     tokenizer = Tokenizer() .setInputCols(["sentence"]) .setOutputCol("token")
6     word_embeddings = WordEmbeddingsModel.pretrained("embeddings_clinical", "en", "clinical/
          ↪ models") .setInputCols(["sentence", "token"]) .setOutputCol("embeddings")
7     clinical_ner = MedicalNerModel.pretrained("ner_clinical", "en", "clinical/models") .
          ↪ setInputCols(["sentence", "token", "embeddings"]) .setOutputCol("ner")
8     ner_converter = NerConverter() .setInputCols(["sentence", "token", "ner"]) .setOutputCol("
          ↪ ner_chunk")
9     clinical_assertion = AssertionDLModel.pretrained("assertion_dl", "en", "clinical/models") .
          ↪ setInputCols(["sentence", "ner_chunk", "embeddings"]) .setOutputCol("assertion")
10
11    nlpPipeline = Pipeline(stages=[
12       documentAssembler,
13       sentenceDetector,
```

```
14        tokenizer,
15        word_embeddings,
16        clinical_ner,
17        ner_converter,
18        clinical_assertion
19    ])
20
21    empty_data = spark.createDataFrame([[””]]).toDF(”text”)
22    model = nlpPipeline.fit(empty_data)
23    return model
```

Os outros modelos aplicados nesta dissertação são carregados de igual forma e todos possuem *outputs* que não são visualmente agradáveis ao utilizador e o seu tratamento será explicado posteriormente.

4.1.3.2 Aplicação dos Modelos

Depois dos modelos carregados, são necessárias funções auxiliares para aplicá-los às notas clínicas que queremos analisar. Seguindo o exemplo do modelo acima, neste caso é efetuado um pequeno pré-processamento da nota clínica para ”limpar”a mesma, retirando todos os *'\n'* presentes. Após isso é feita a chamada da função para anotar o texto e é construído o *Pandas DataFrame* que contém os resultados da anotação.

Lista de Tabelas 4.5: Função de aplicação do modelo de *Clinical Assertion*

```
1  def get_assertion_results(text):
2
3     model = load_clinical_assertion()
4     sem_n=text.replace(r'\n',' ')
5     light_model = LightPipeline(model)
6     light_result = light_model.fullAnnotate(sem_n)[0]
7     chunks=[]
8     entities=[]
9     status=[]
10
11    for n,m in zip(light_result['ner_chunk'],light_result['assertion']):
12
13       chunks.append(n.result)
14       entities.append(n.metadata['entity'])
15       status.append(m.result)
16
17    df = pd.DataFrame({'chunks':chunks, 'entities':entities, 'assertion':status})
18
19    return df
```

No caso do modelo de resolução de ICD-10, o procedimento é semelhante.

4.1.3.3 Tratamento dos Resultados

Todos os modelos utilizados nesta aplicação, apresentam *outputs* que, do ponto de vista do utilizador, são irrelevantes. Isto prende-se ao facto destes não serem visualmente apelativos do ponto de vista de um utilizador que seja um profissional de saúde. Por exemplo, na análise de um médico ao historial de um paciente, a apresentação de uma tabela apenas com os termos médicos e o resultado da *assertion* não iria permitir uma análise contextual das doenças, problemas ou tratamentos presentes e/ou ausentes.

Para ultrapassar este problema, a etapa seguinte passou pelo tratamento de cada *output* de forma a integrá-lo no que seria uma aplicação para utilizar um profissional de saúde. As resoluções efetuadas para cada modelo foram:

- **Spell Checker**: o resultado da aplicação deste modelo numa nota clínica é uma lista que contém uma lista de tokens. A resolução passou pela utilização de funções simples que transformavam de volta esses tokens todos na string original, mas corrigida.

Lista de Tabelas 4.6: Exemplo de tratamento do output do modelo de *Spell Checking*

```
c = apply_spellcheck(texto)
txt = sum(c.values(),[])
      txt = " ".join(str(x) for x in txt)
```

- **Clinical Assertion**: o *output* deste modelo era uma tabela que apresentava os termos médicos do texto e o resultado (presente, ausente e hipotético). Em notas clínicas com maior extensão, ou nas mais pequenas, a apresentação de uma tabela ao médico com estes resultados não seria visualmente muito útil pois todos os termos estariam fora de contexto e não permitiriam ao médico analisar o paciente com maior eficácia, comparativamente a uma nota clínica sem qualquer tratamento. Para solucionar este problema e providenciar ao profissional de saúde uma melhor análise a nota clínica, é efetuado um *highlighting* dos termos médicos identificados na nota clínica que é impressa por baixo da tabela de resultados. Este processo inicia-se com a criação de uma lista de tuplos a partir do *output* que contém o termo médico identificado juntamente com o resultado da *assertion* atribuído a esse termo. Com essa lista, a nota clínica original é percorrida através de um ciclo para que, sempre que encontrar um termo médico pertencente à lista de tuplos, verifique o seu resultado e aplique uma *mark* personalizada consoante o mesmo. Assim, o texto é impresso na aplicação com um visual mais apelativo aos olhos de um profissional de saúde.

Lista de Tabelas 4.7: Processo de *highlighting* no modelo de *Clinical Assertion*

```
lista_combinada = x.values.tolist()
      lista_tuplos = []
```

```
    for i in lista_combinada:
        tuplo= tuple(i)
        lista_tuplos.append(tuplo)
    for i in lista_tuplos:
        if i[1] == 'present':
            texto = texto.replace(i[0], "<mark class=\"orange\">" + i[0] + "</mark>")
        if i[1] == 'absent':
            texto = texto.replace(i[0], "<mark class=\"green\">" + i[0] + "</mark>")
        if i[1] == 'hypothetical':
            texto = texto.replace(i[0], "<mark>" + i[0] + "</mark>")
```

- **Clinical Entity Recognition**: assim como no modelo enunciado anteriormente, o *output* deste é também uma tabela. Como tal, o tratamento efetuado foi semelhante. A partir do resultado é criada uma lista de tuplos que, neste caso, contém no primeiro membro o termo médico e, no segundo membro, o código ICD-10 detetado pelo modelo. Desta maneira, na nota clínica impressa por baixo da tabela, foi criado, com recurso ao HTML e ao CSS, um estilo em que o médico pode passar o rato por cima de um termo médico e o respetivo código desse termo aparecerá por cima.

Lista de Tabelas 4.8: Processo de *highlighting* no modelo de *Clinical ICD-10*

```
lista_combinada = y.values.tolist()
    lista_tuplos = []
    for i in lista_combinada:
        tuplo= tuple(i)
        lista_tuplos.append(tuplo)
    for i in lista_tuplos:
        texto = texto.replace(i[0], ("<span class=\"hovertext\" data-hover=\"" + i[1] +
            ↪ "\">" + i[0] + "</span>"))
```

É também importante referir que estes 2 últimos modelos, possuem nas suas tabelas colunas irrelevantes para o objetivo deste trabalho que são eliminadas do *DataFrame* resultante para facilitar o uso da aplicação por parte do utilizador.

4.1.4 *Routing*

As rotas da aplicação, apesar de estarem todas no mesmo ficheiro, podem-se considerar divididas por várias secções.

- **Autenticação:** esta secção contém as rotas relacionadas com a autenticação de um profissional de saúde: *login*, *sign_up* e *logout*.

- **Carregamento de Notas Clínicas:** nesta secção existem 2 rotas primárias: o carregamento de um ficheiro e o carregamento em formato texto. Estas rotas encontram-se separadas por funcionarem de formas diferentes, quer quando são guardadas, quer depois para utilização das mesmas.
- **Anotação Clínica:** a maior secção do trabalho. Nesta secção consideram-se 6 rotas, 3 para cada modelo por tipo de nota clínica. É nestas rotas que é chamado o carregamento de um modelo, é feita a sua aplicação e o tratamento dos resultados.

4.2 Interface

A primeira página com a qual o utilizador tem contacto é a *homepage*, que aparece sempre que a aplicação é iniciada. Nesta são apresentadas as principais características da aplicação e um pequeno resumo da mesma. O utilizador pode então aceder à zona de autenticação para depois então poder aceder à lista de notas clínicas ou efetuar o *upload* de uma nota clínica (Figura 27).

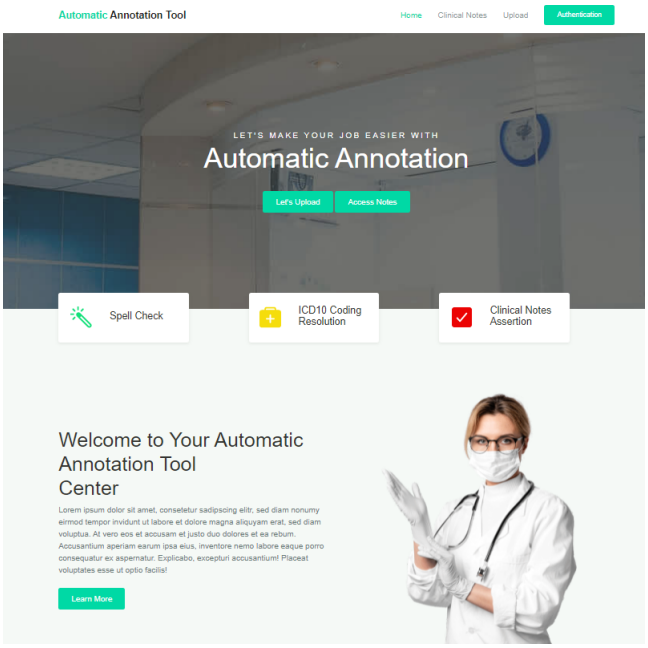


Figura 27: *Homepage* da aplicação.

Na área da autenticação, um profissional de saúde poderá então efetuar o *login* (Figura 28) ou, caso não possua conta, registar a sua conta de profissional de saúde (Figura 29).

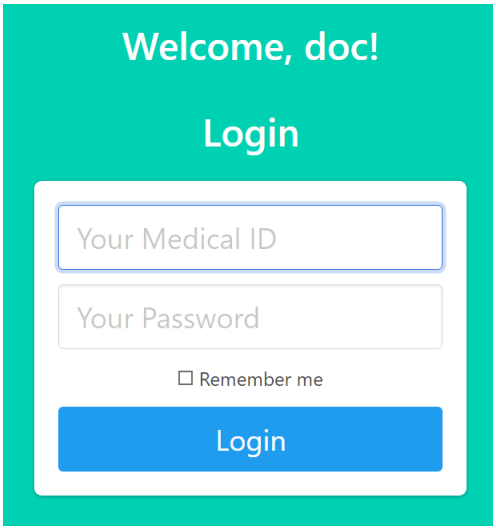


Figura 28: Interface para o *login* na aplicação.

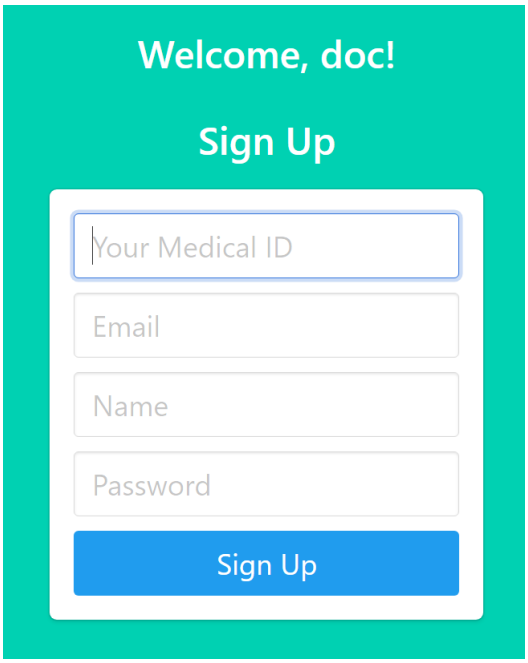


Figura 29: Interface para efetuar o registo na aplicação.

Após a autenticação, o profissional de saúde pode então aceder a um conjunto de notas clínicas da base de dados definido por ele mesmo (Figura 31) que serão apresentadas em forma de tabela com botões de acesso individual (Figura 32) ou efetuar o *upload* da própria nota clínica que deseja anotar (Figura 30).

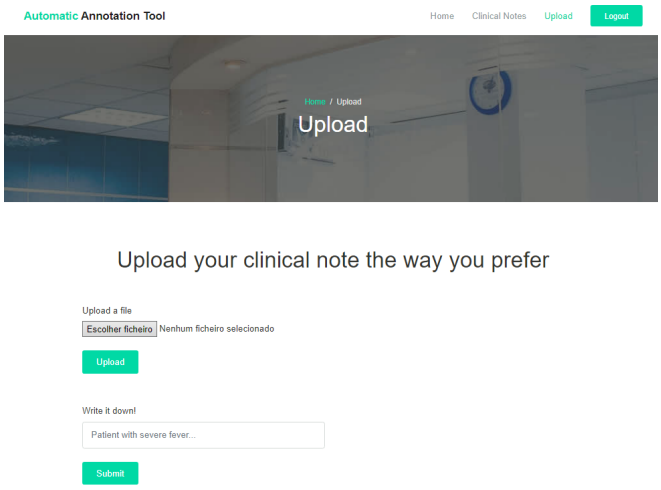


Figura 30: Interface de *upload* da aplicação.

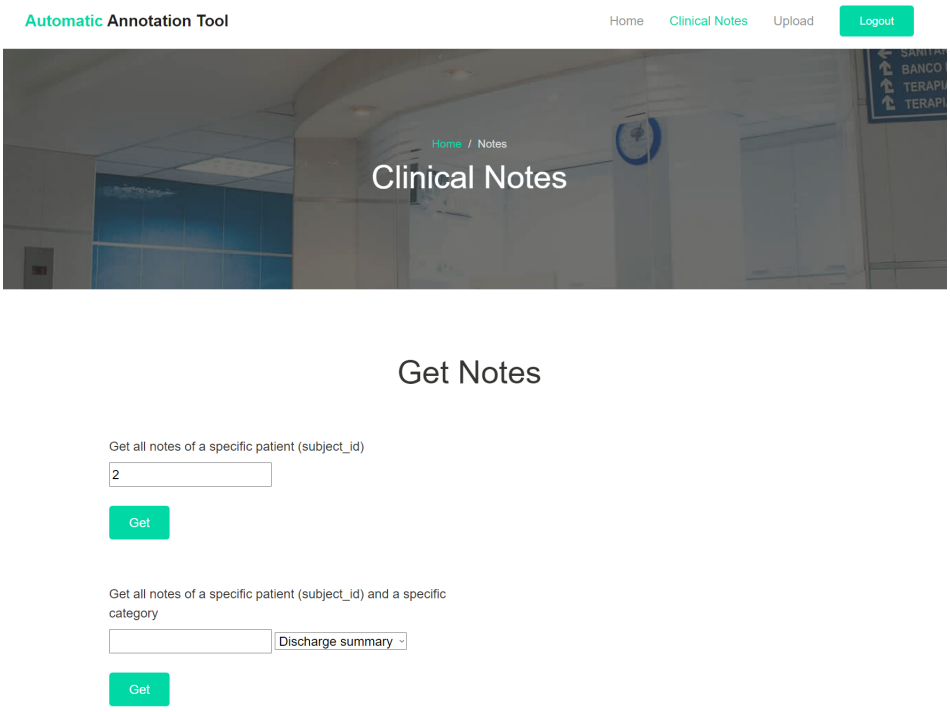


Figura 31: Interface para pesquisar notas clínicas.

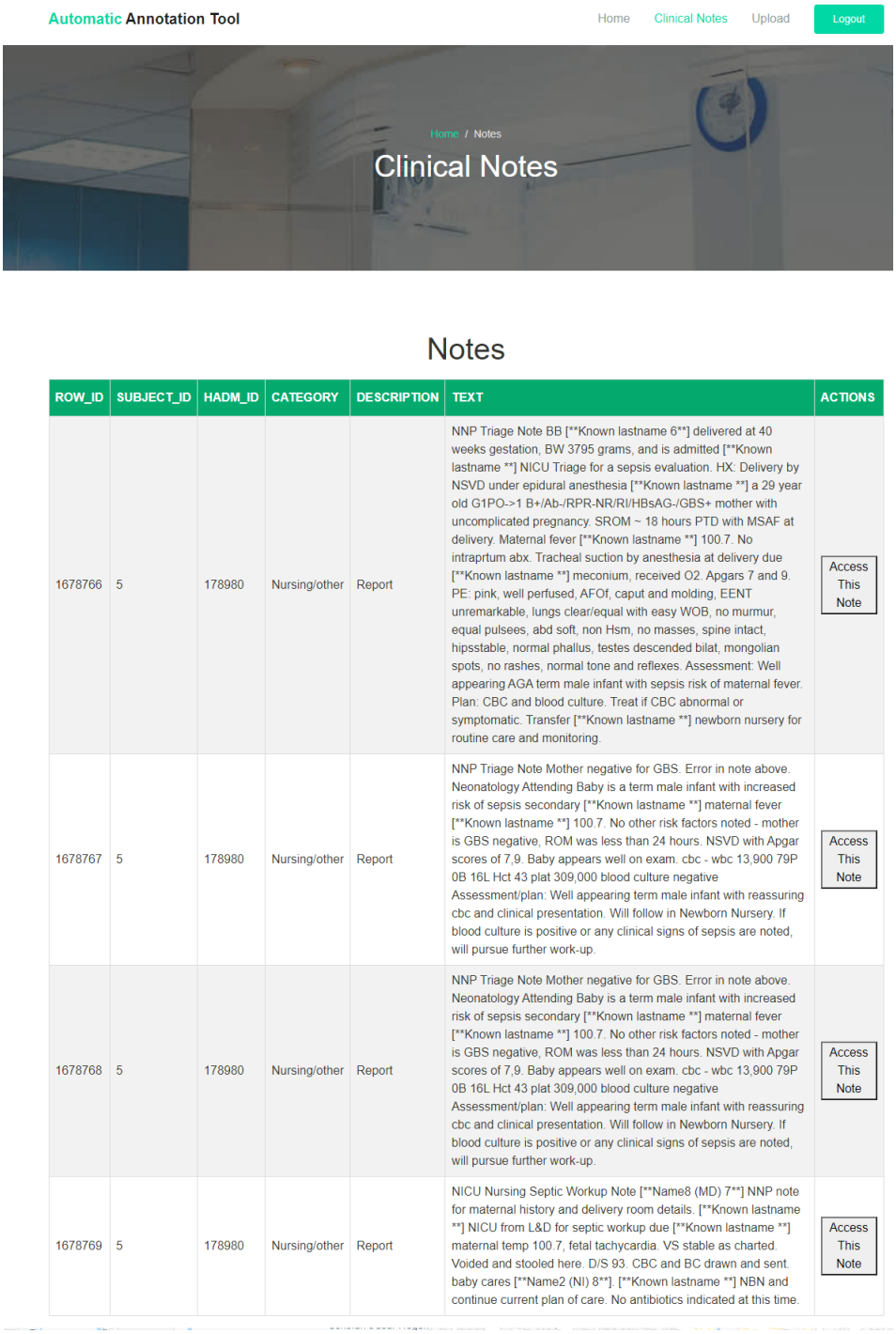


Figura 32: Lista de notas clínicas na aplicação.

Sempre que o utilizador aceder a apenas uma nota clínica individual, quer esta seja proveniente da base de dados ou das mãos do profissional de saúde, terá à sua disposição 3 botões com os nomes dos 3 modelos possíveis de aplicar na mesma (Figura 33).

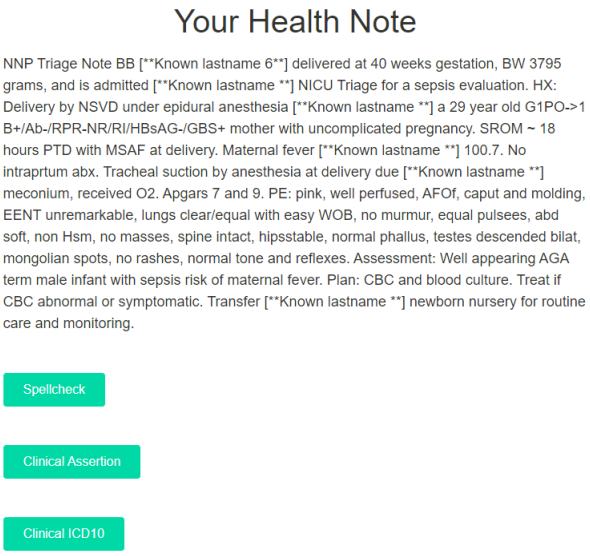
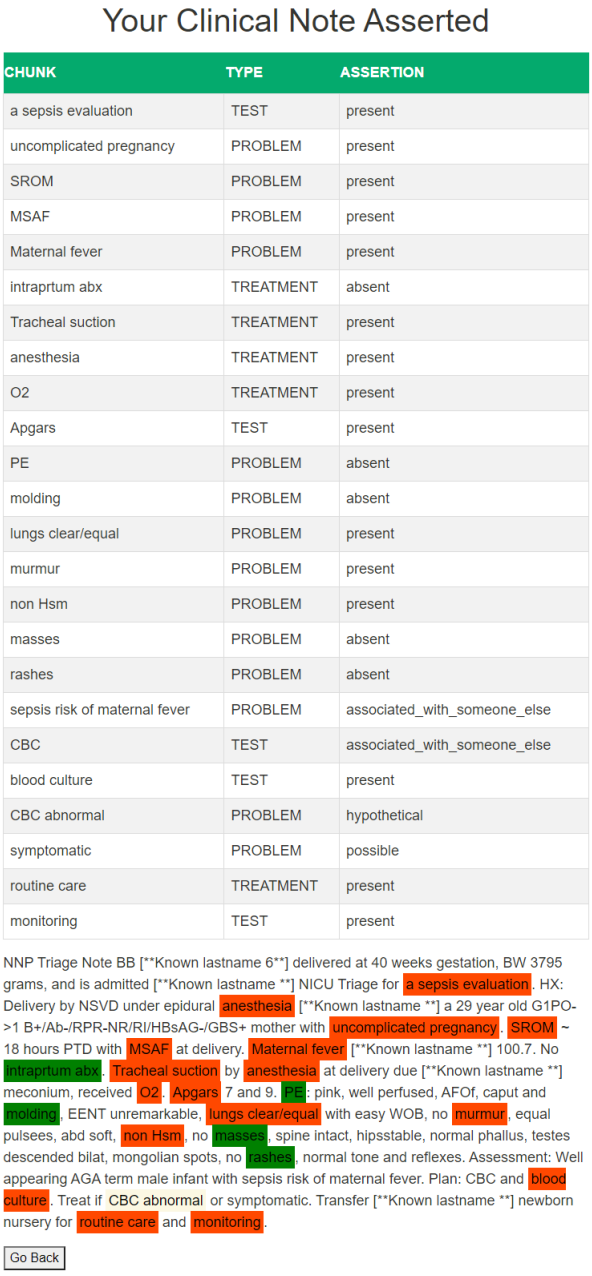


Figura 33: Vista de uma nota individual na aplicação.

No caso de aplicação do modelo de *Clinical Assertion*, o utilizador poderá então visualizar a tabela e a nota clínica personalizada referentes à nota clínica que acedeu (Figura 34).



Your Clinical Note Asserted

CHUNK	TYPE	ASSERTION
a sepsis evaluation	TEST	present
uncomplicated pregnancy	PROBLEM	present
SROM	PROBLEM	present
MSAF	PROBLEM	present
Maternal fever	PROBLEM	present
intraprtum abx	TREATMENT	absent
Tracheal suction	TREATMENT	present
anesthesia	TREATMENT	present
O2	TREATMENT	present
Apgars	TEST	present
PE	PROBLEM	absent
molding	PROBLEM	absent
lungs clear/equal	PROBLEM	present
murmur	PROBLEM	present
non Hsm	PROBLEM	present
masses	PROBLEM	absent
rashes	PROBLEM	absent
sepsis risk of maternal fever	PROBLEM	associated_with_someone_else
CBC	TEST	associated_with_someone_else
blood culture	TEST	present
CBC abnormal	PROBLEM	hypothetical
symptomatic	PROBLEM	possible
routine care	TREATMENT	present
monitoring	TEST	present

NNP Triage Note BB [**Known lastname 6**] delivered at 40 weeks gestation, BW 3795 grams, and is admitted [**Known lastname **] NICU Triage for a sepsis evaluation. HX: Delivery by NSVD under epidural anesthesia [**Known lastname **] a 29 year old G1PO->1 B+/Ab-/RPR-NR/RI/HBsAG-/GBS+ mother with uncomplicated pregnancy, SROM ~ 18 hours PTD with MSAF at delivery. Maternal fever [**Known lastname **] 100.7. No intraprtum abx. Tracheal suction by anesthesia at delivery due [**Known lastname **] meconium, received O2. Apgars 7 and 9. PE: pink, well perfused, AFOf, caput and molding, EENT unremarkable, lungs clear/equal with easy WOB, no murmur, equal pulsees, abd soft, non Hsm, no masses, spine intact, hipsstable, normal phallus, testes descended bilat, mongolian spots, no rashes, normal tone and reflexes. Assessment: Well appearing AGA term male infant with sepsis risk of maternal fever. Plan: CBC and blood culture. Treat if CBC abnormal or symptomatic. Transfer [**Known lastname **] newborn nursery for routine care and monitoring.

Go Back

Figura 34: Resultado e vista da aplicação de *assertion* na aplicação.

Caso a opção passe por aplicar o modelo de resolução ICD-10 então o utilizador poderá visualizar a tabela resultante e a nota clínica corretamente sublinhada. Nos termos médicos sublinhados, o utilizador poderá colocar o rato por cima e o respetivo código ICD-10 irá aparecer (Figura 35).

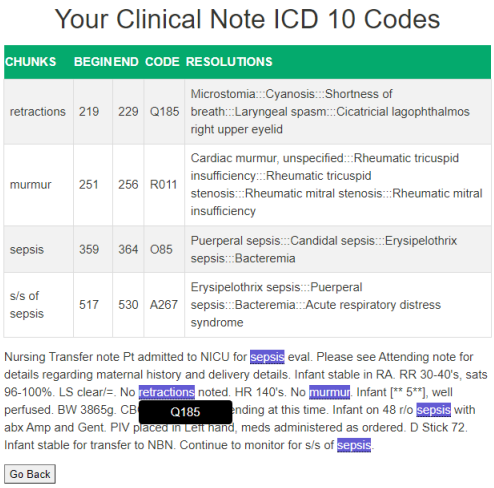


Figura 35: Resultado e vista da resolução de uma nota clínica ao nível da codificação ICD-10.

4.3 Use Case

4.3.1 Aplicação de um Modelo a uma Nota Clínica da Base de Dados

O *use case* mais típico desta aplicação é fazer a anotação de uma nota clínica de um paciente. Supondo que o profissional de saúde está autenticado na aplicação e se encontra na página principal (Figura 27) deve selecionar a opção *Access Notes* na parte central da página ou *Clinical Notes* na barra de navegação em cima. Quando estiver no ecrã que permite pesquisar notas (Figura 31) cabe ao utilizador decidir quais pretende aceder. O utilizador pode aceder a todas as notas clínicas de um paciente, especificando o seu *subject_id* ou então pode aceder às notas de uma categoria específica de um paciente específico, especificando o *subject_id* e a *category* da nota clínica.

Quando o utilizador se encontrar no ecrã que apresenta as notas clínicas cujos atributos verificaram os valores selecionados (Figura 32), poderá então aceder a cada uma das notas de forma individual onde poderá escolher aplicar um dos 3 modelos disponíveis (Figura 33).

Sempre que o utilizador aplicar um modelo, pode facilmente voltar atrás e aplicar outro modelo, ou voltar a escolher uma nota para aceder individualmente.

É importante salientar que o tempo de espera de aplicação dos modelos, que inclui o carregamento e a anotação, apresenta um valor maior na 1ª utilização comparativamente às restantes. Isto deve-se ao facto de, após a primeira aplicação, os modelos serem guardados em *cache* pelo que o uso continuado da aplicação não oferece problemas. Além disso, estes modelos oferecem a possibilidade de serem guardados localmente no computador passando o seu carregamento a ser feito de forma *offline*.

Capítulo 5

Prova de Conceito e Avaliação

Neste capítulo é realizada uma Análise SWOT à aplicação apresentada. Após esta, é descrito todo o processo de testagem dos modelos utilizados na aplicação bem como a metodologia seguida.

5.1 Análise SWOT

Uma análise SWOT é uma ferramenta de identificação das forças, oportunidades, fraquezas e ameaças do objeto de estudo. Através de uma análise simples, mas rigorosa, é possível estruturar de forma estratégica e, como o nome indica, maximizar oportunidades oriundas de fatores externos, através das forças internas da aplicação e minimizar as ameaças de fatores externos, tendo em conta as fraquezas internas da aplicação. Existem 2 tipos de ambientes que influenciam a análise: externo e interno. O ambiente interno refere-se às características, neste caso, da aplicação, enquanto que o ambiente externo se refere às características do mercado onde esta se insere [54].

Figura 36: Matriz da Análise SWOT.

Para esta aplicação em específico, resultou a seguinte análise SWOT:

- **Forças:**
  - Escalabilidade;
  - Elevado índice de usabilidade;
  - Anotação automática de informação clínica;
  - Rápida análise de pacientes;
  - Diminuição da probabilidade de erros na análise por parte do profissional de saúde.
- **Fraquezas:**
  - Requer conectividade à *internet* no mínimo, uma vez;
  - Manuseamento de dados pessoais;
  - Velocidades de carregamento dependem da qualidade da ligação à *internet*;
- **Oportunidades:**
  - Possibilidade de acrescentar uma maior quantidade e variedade de modelos para anotação;
  - Redução da carga de trabalho dos profissionais de saúde;
  - Aumento da capacidade física e psicológica dos profissionais de saúde;
  - Constante desenvolvimento e evolução do PLN e do *Spark NLP* possibilita uma aplicação sempre atualizada e em constante evolução;

  - Possibilidade de aumentar a velocidade de trabalho nos hospitais, consoante adoção da aplicação.

- **Ameaças:**
  - Inexistência de dados portugueses;
  - Utilização de registos eletrónicos estruturados já com anotação que tornem desnecessária a aplicação.

5.2 Método de Teste

Um dos maiores objetivos do treino de modelos de PLN passa pela generalização. A testagem de modelos além de custosa em termos de recursos, muito raramente permite fazer rápidas iterações. Além disso, muitas das vezes o que acontece é que o *dataset* utilizado para teste é semelhante ao utilizado em treino levando a que os resultados sejam claramente favoráveis, não avaliando, de forma correta, o funcionamento desse modelo em contexto de vida real. Normalmente, a avaliação de performance incide apenas em 1 ou 2 estatísticas tornando assim muito complicado perceber onde o modelo está a falhar e qual a melhor maneira de o corrigir [55].

Em 2019, num artigo publicado por um grupo de investigadores da universidade de *Washington*, concluíram que as análises de erros de modelos eram trabalhosas e subjetivas. Para dar a volta a este problema identificaram 3 princípios necessários para uma análise de erros ser bem sucedida [56]:

- Construção de blocos de código precisos e montáveis na linguagem de domínio;
- A análise deve ser escalada de forma simples substituindo blocos de código de modo a cobrir todos os sucessos ou falhanços relevantes;
- Deve ser possível reescrever blocos de código e/ou regras de forma a aumentar a replicação de dados e permitir testar vários modelos para um mesmo grupo de regras.

Neste trabalho foi desenvolvida uma ferramenta que permite aplicar todos os princípios identificados pelos autores.

Já em 2020, 2 dos autores do artigo anterior, juntamente com mais 2 outros investigadores, publicaram um artigo que propôs uma nova metodologia de avaliação de modelos acompanhada de uma nova ferramenta, de seu nome *CheckList*, focada então na avaliação comportamental de modelos de PLN.

Este artigo guia os utilizadores para o que deve ser testado, providenciando uma lista de testes linguísticos que cobrem todos os potencias pontos de falha no comportamento do modelo a ser testado. Esta ferramenta além de especificar o que deve ser testado, permite, com a sua biblioteca, que os utilizadores

criem uma enorme quantidade de casos de testes facilmente. Comparativamente falando, um *benchmark* tradicional diz-nos que um modelo é tão preciso quanto um humano. O *benchmark* do *CheckList* revela-nos a variedade de *bugs* existentes nas capacidades linguísticas do modelo.

Para um utilizador fazer *CheckList* do seu modelo, preenche as células de uma matriz, cada célula significando uma capacidade por tipo de teste.

Esta metodologia encoraja os utilizadores a perceber como as diferentes capacidades se manifestam em cada tarefa e a criar diferentes tipos de teste para cada uma destas.

As capacidades sugeridas pelos autores são:

- ***Vocabulary + POS:*** grupos de palavras importantes para a tarefa em questão;
- ***Taxonomy:*** com a substituição de partes dos dados por sinónimos, antónimos, etc;
- ***Robustness:*** adicionado erros gramaticais aos dados;
- ***Named Entity Recognition (NER):*** alterar entidades;
- ***Fairness:*** a capacidade de um modelo funcionar igualmente para todos os grupos de pessoas a respeito de certos atributos. Por exemplo, o modelo deve prever corretamente a mesma *label* para todos os géneros.
- ***Temporal Understanding:*** alterações temporais no contexto da frase;
- ***Negation:*** tornar a frase negativa ou positiva;
- ***Coreference:*** alteração de expressões por outras que se refiram à mesma entidade numa frase;
- ***Semantic Role Labeling (SRL):*** compreensão das funções semânticas presentes na frase;
- ***Logic:*** alterar a lógica gramatical da frase.

Para cada capacidade, existem 3 testes a fazer:

- ***Minimum Functionality Tests:*** estes são inspirados nos testes unitários de engenharia de *software* e são uma coleção de simples exemplos para perceber o comportamento do modelo para cada tipo de capacidade. É particularmente útil para perceber quando é que os modelos utilizam atalhos para resolver problemas mais completos.
- ***Invariance:*** neste tipo de testes, são aplicadas perturbações nos dados que preservam a *label* destes. São aplicadas diferentes funções de perturbação, como por exemplo a troca de palavras por sinónimos ou antónimos e a adição de erros gramaticais.

- ***Directional Expectation Tests:*** estes são similares aos anteriores, porém nestes é esperado que a *label* atribuída seja alterada.

Para além da criatividade necessária para a criação de dados de teste, normalmente estes acabam sempre por ser de baixa qualidade e em pouca quantidade. A *CheckList* providencia ao utilizador ferramentas de criação de dados de teste e funções pré-definidas de perturbação de dados [57].

Assim, o método de teste que foi aplicado nos 2 modelos principais desta aplicação (*assertion* e *ICD-10 resolution*) foi este mesmo. Para isso foram definidas 2 matrizes de testes onde foram decididos os testes necessários a fazer tendo em conta as tarefas de cada modelo.

Por exemplo, em nenhum dos modelos fazia sentido testar a capacidade *Temporal Understanding* viste que estes modelos detetam termos médicos independentemente do espaço temporal onde estes se encontrem.

A nível de *sample* utilizado para os testes foram utilizados 2 grupos de frases, selecionados sempre aleatoriamente do conjunto completo de códigos ICD-10, sendo um grupo de 10 frases e outro de 100 frases. Tendo em conta os recursos temporais e de processamento necessários, fazer testes com mais de 100 frases tornar-se-ia complicado sem ter acesso a uma máquina bastante superior. O grupo de 10 frases serviu para os testes mais simples nos quais conseguimos concluir que o teste é sempre bem sucedido, enquanto que o de 100 frases foi necessário para testes mais completos e uma melhor análise.

5.2.1 *Clinical Assertion*

Para a análise deste modelo as capacidades que foram testadas, juntamente com os tipos de teste foram:

- ***Robustness x INV:*** testar a preservação das *labels* quando os dados são alterados com erros gramaticais ou abreviação de palavras.
- ***Negation x MFT x INV x DIR:*** esta é a capacidade que dá o nome ao modelo, como tal, foram efetuados os 3 tipos de teste nesta capacidade: os testes simples, os de preservação da *label* e os de alteração da *label*. Além disso, devido às diferentes possibilidades de negação é também testada alguma taxonomia do modelo.

5.2.2 *Clinical ICD-10*

Quanto à resolução de códigos ICD-10 os tipos de testes efetuados por capacidades foram:

- ***Taxonomy x INV:*** na resolução de códigos ICD-10 é importante testar a preservação da *label* quando palavras são trocadas por sinónimos.

- ***NER x MFT x INV x DIR e Resolvers:*** neste secção além do teste da capacidade são testados, ao mesmo tempo, os *Resolvers*. O *Resolver* é o responsável pela resolução/atribuição de um código ICD-10 à entidade detetada. A deteção da entidade é a capacidade necessária antes da atribuição de um código ICD-10 e uma das etapas principais do modelo logo era importante testar a capacidade na forma de testes simples e nos testes de preservação das *labels*. Além disso é importante testar a alteração de *label* caso se substitua uma doença numa frase por outra qualquer.

- ***Robustness x INV :*** testar a preservação das *labels* quando os dados são alterados com erros gramaticais ou abreviação de palavras.

5.3 Resultados

5.3.1 *Clinical Assertion*

Relativamente aos testes deste modelo a tarefa é mais complexa. No modelo anterior, os próprios códigos que queremos detetar possuem frases que os descrevem, tornando relativamente fácil a aplicação do modelo porque os dados de teste são, no fundo, os códigos presentes na terminologia e as frases já estão todas construídas sendo apenas necessária a sua perturbação consoante o teste. Este modelo, como procura detetar se um termo médico está presente ou ausente de uma nota clínica apresenta a dificuldade da quantidade de dados de teste. A resolução deste obstáculo passou pela utilização da própria aplicação para captar frases de teste.

5.3.1.1 *Negation*

Esta é a capacidade principal deste modelo. É nesta que reside a objetividade deste modelo e, portanto, perceber os resultados dos testes nesta capacidade é determinante. Este processo começou então pela seleção de 10 a 15 frases que apresentassem a *label present* para o termo médico contido na frase. Mais uma vez, é de salientar que estes modelos estão treinados em contextos de notas clínicas e não de frases isoladas.

Lista de Tabelas 5.1: Frases para os testes simples do modelo *Clinical Assertion*

```
examples = [('present', 'admitted to NICU for sepsis eval'),
        ('present', 'PIV placed in left hand'),
        ('present', 'meds administered as ordered'),
        ('present', 'male infant with increased risk of sepsis'),
        ('present', 'Mother was treated with antibiotics'),
        ('present', 'he presented again to the [**Hospital1 346**] after being found to have
            ↪ a systolic blood pressure'),
        ('present', 'He was resuscitated with epinephrine'),
```

```
        ('present', 'He was also given ProMod shakes'),
        ('present', 'The patient was transfused one unit of packed red blood cells'),
        ('present', 'Well appearing term male infant with reassuring cbc'),
        ('present', 'The patient is a 65 year-old woman with end stage renal disease'),
        ('present', 'The ultrasound demonstrated that there was no'),
        ('present', 'She went home with services on the following medications'),
        ('present', '65-year-old woman status post renal transplant')]
```

Neste teste simples, os resultados foram bastantes satisfatórios para os diferentes tipos de frases (Figura 37).

```
In [48]: suite.summary()

         Negation - MFT

         MFT 0
         Test cases:      14
         Fails (rate):    1 (7.1%)

         Example fails:
         no_assert_detected Well appearing term male infant with reassuring cbc
         ----
```

Figura 37: Resultados do teste simples da capacidade de negação.

A inserção de negatividade das frases foi feita através de uma função da biblioteca *SpaCy*. A utilização desta biblioteca concorrente do *Spark NLP* deve-se ao facto da ferramenta *CheckList* estar integrada e possuir funções úteis para a criação de dados de teste. Esta função apesar de adicionar negatividade em todas as frases, não foi suficiente, visto que nestes casos queríamos testar a ausência de um termo médico e não a negação da frase por completo.

Lista de Tabelas 5.2: Inserção de negatividade nas frases de teste do modelo de *Clinical Assertion*

```
%%%
import spacy
nlp = spacy.load('en_core_web_sm')
%%%
for t in phrases:
   print(t)
   print(Perturb.add_negation(nlp(t)))
   print()
%%%

negated = [('absent','admitted to NICU but didnt need a sepsis eval'),
         ('absent','PIV not placed in left hand'),
        ('absent','meds were not administered as ordered'),
         ('absent','male infant without increased risk of sepsis'),
         ('absent','Mother was treated without antibiotics'),
         ('absent','he presented again to the [**Hospital1 346**] although not found to have a
             ↪ systolic blood pressure'),
```

```
17          ('absent','He was resuscitated without epinephrine'),
18          ('absent','He was also not given ProMod shakes'),
19          ('absent','The patient was not transfused one unit of packed red blood cells'),
20          ('absent','Well appearing term male infant without reassuring cbc'),
21          ('absent','The patient is a 65 year-old woman without end stage renal disease'),
22          ('absent','The ultrasound was not done'),
23          ('absent','She went home with services and did not take following medications'),
24          ('absent','65-year-old woman did not need a renal transplant')]
```

Os resultados neste *sample* foram os esperados. O modelo utilizado foi treinado em notas clínicas pelo que os seus resultados em frases independentes de contexto nunca iriam ser perfeitos. Apesar de o modelo ter falhado praticamente metade dos casos, é importante analisar as frases e perceber que algumas, de facto, não evidenciam com total certeza a ausência do termo médico (Figura 38).

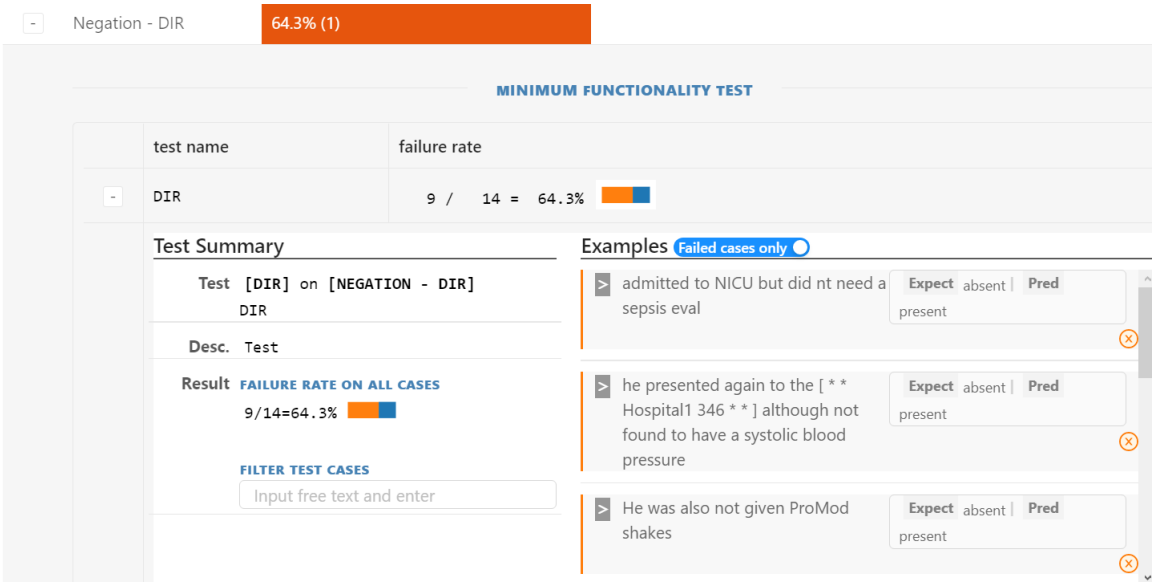


Figura 38: Resultados do teste de mudança de *label* da capacidade de negação.

5.3.2 *Clinical ICD-10*

O processo para testar este modelo começou, inicialmente, pela análise da listagem de códigos ICD-10 existentes na área. Era necessário perceber como eram construídas as frases, qual a sua extensão, quantos códigos existiam e outras características que iriam definir o modo como os testes iam ser feitos.

Um código ICD-10 é constituído por 1 letra inicial de A até Z seguida de números e podendo finalizar ou não com outra letra (Figuras 40 e 41). Estes códigos estão organizados por grandes grupos de doenças correspondentes à primeira letra que inicia o código (Figura 39). Um dado importante sobre o significado de cada código e que tem bastante impacto nos testes é que estes códigos podem ter vários sinónimos (Figura 42).

2022 ICD-10-CM Codes

- A00-B99 Certain infectious and parasitic diseases
- C00-D49 Neoplasms
- D50-D89 Diseases of the blood and blood-forming organs and certain disorders involving the immune mechanism
- E00-E89 Endocrine, nutritional and metabolic diseases
- F01-F99 Mental, Behavioral and Neurodevelopmental disorders
- G00-G99 Diseases of the nervous system
- H00-H59 Diseases of the eye and adnexa
- H60-H95 Diseases of the ear and mastoid process
- I00-I99 Diseases of the circulatory system
- J00-J99 Diseases of the respiratory system
- K00-K95 Diseases of the digestive system
- L00-L99 Diseases of the skin and subcutaneous tissue
- M00-M99 Diseases of the musculoskeletal system and connective tissue
- N00-N99 Diseases of the genitourinary system
- O00-O9A Pregnancy, childbirth and the puerperium
- P00-P96 Certain conditions originating in the perinatal period
- Q00-Q99 Congenital malformations, deformations and chromosomal abnormalities
- R00-R99 Symptoms, signs and abnormal clinical and laboratory findings, not elsewhere classified
- S00-T88 Injury, poisoning and certain other consequences of external causes
- U00-U85 Codes for special purposes
- V00-Y99 External causes of morbidity
- Z00-Z99 Factors influencing health status and contact with health services

Figura 39: Listagem de códigos existentes na nomenclatura ICD-10-CM.

2022 ICD-10-CM Diagnosis Code G20

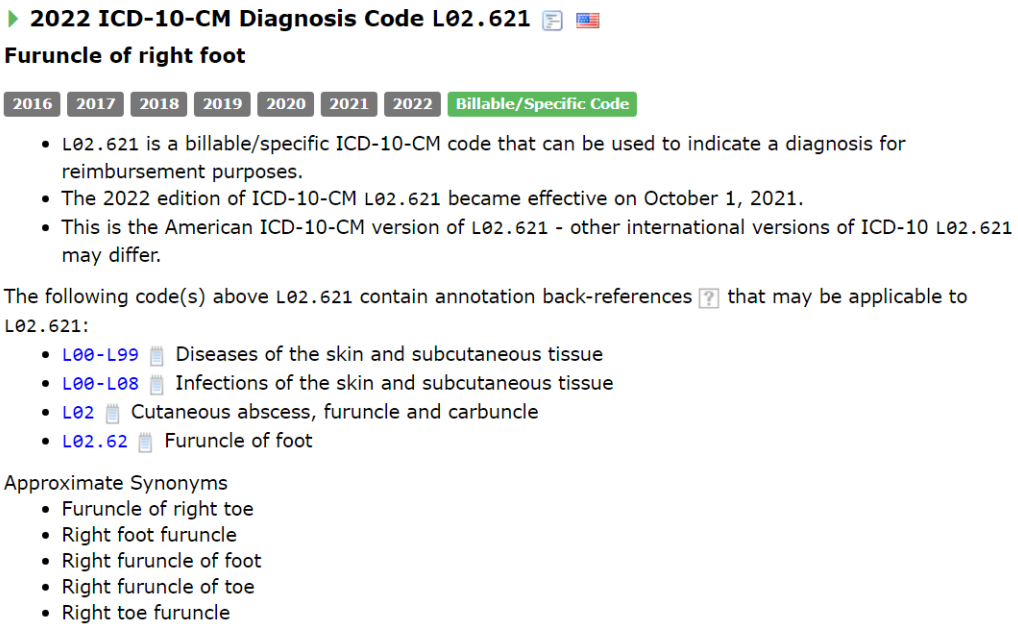
Parkinson's disease

Figura 40: Exemplo de um código mais simples.

2022 ICD-10-CM Diagnosis Code E11.3549

Type 2 diabetes mellitus with proliferative diabetic retinopathy with combined traction retinal detachment and rhegmatogenous retinal detachment, unspecified eye

Figura 41: Exemplo de um código mais extenso.



2022 ICD-10-CM Diagnosis Code L02.621

Furuncle of right foot

2016 2017 2018 2019 2020 2021 2022 Billable/Specific Code

- L02.621 is a billable/specific ICD-10-CM code that can be used to indicate a diagnosis for reimbursement purposes.
- The 2022 edition of ICD-10-CM L02.621 became effective on October 1, 2021.
- This is the American ICD-10-CM version of L02.621 - other international versions of ICD-10 L02.621 may differ.

The following code(s) above L02.621 contain annotation back-references that may be applicable to L02.621:

- L00-L99 Diseases of the skin and subcutaneous tissue
- L00-L08 Infections of the skin and subcutaneous tissue
- L02 Cutaneous abscess, furuncle and carbuncle
- L02.62 Furuncle of foot

Approximate Synonyms

- Furuncle of right toe
- Right foot furuncle
- Right furuncle of foot
- Right furuncle of toe
- Right toe furuncle

Figura 42: Exemplo de um código e dos seus sinónimos.

5.3.2.1 *Taxonomy*

Neste teste o objetivo era perceber a influência que a troca de palavras das frases por sinónimos poderia ter na eficácia do modelo. Para este teste então foi utilizada uma função que trocava os conectores das frases por sinónimos. Assim, a frase continuaria com o mesmo sentido mas a gramática que relacionava os termos da frase estava diferente.

Lista de Tabelas 5.3: Processo de perturbação das frases originais e adição do teste INV - *Typos*

```
def inv_connectors(x, *args, **kwargs):
   # Returns empty or a list of strings with phrase connector changed - SYNONYMS FOR DUE TO
   list_of_terms = ['due to','d/t', 'secondary to', '2/2' , 'related to', 'r/t', 'in']
   ret = []
   for t in list_of_terms:
      if re.search(r'\b%s\b' % t, x):
         ret.extend([re.sub(r'\b%s\b' % t, t2, x) for t2 in list_of_terms if t != t2])
   return ret

ret_inv_connectors = Perturb.perturb(phrases, inv_connectors, keep_original=True)
```

Sendo assim, foram selecionados 2 exemplos de códigos ICD-10 cujo significado era uma frase com uma relação.

Lista de Tabelas 5.4: Frases-teste e as suas perturbações

```
examples = [('N4602', 'Azoospermia due to extratesticular causes'),
        ('P57', 'Kernicterus due to isoimmunization')]
```

```
['Kernicterus due to isoimmunization',
 'Kernicterus due to isoimmunization',
 'Kernicterus d/t isoimmunization',
 'Kernicterus secondary to isoimmunization',
 'Kernicterus 2/2 isoimmunization',
 'Kernicterus related to isoimmunization',
 'Kernicterus r/t isoimmunization',
 'Kernicterus in isoimmunization']

['Azoospermia due to extratesticular causes',
 'Azoospermia d/t extratesticular causes',
 'Azoospermia secondary to extratesticular causes',
 'Azoospermia 2/2 extratesticular causes',
 'Azoospermia related to extratesticular causes',
 'Azoospermia r/t extratesticular causes',
 'Azoospermia in extratesticular causes']
```

Os códigos ICD-10 destas 2 frases são, respetivamente, P57 e N4602. Em ambos os casos o modelo reagiu bem e é possível concluir que a alteração dos conectores não tem praticamente impacto nos resultados do modelo (Figuras 43 e 44). Pela análise feita também se conclui que nas frases em que a *label* não foi preservada, o modelo apenas errou na especificação do código, avaliando corretamente a categoria da frase.

Figura 43: Resultado do primeiro teste de taxonomia no modelo de Clinical ICD-10.

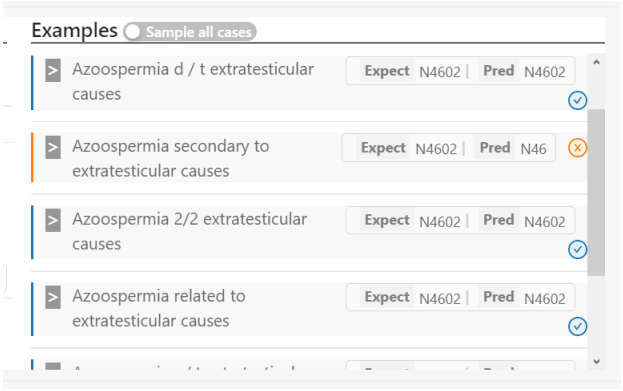


Figura 44: Resultado do segundo teste de taxonomia no modelo de Clinical ICD-10.

É importante ter sempre em conta que neste teste o objetivo é avaliar a preservação da *label* atribuída à frase original comparativamente à frase perturbada. É possível que aconteçam casos em que o modelo avalia uma frase incorretamente relativamente ao código real mas preserva essa *label* quando a frase é perturbada. Neste caso, como o código real é uma especificação do código detetado podemos assumir que o teste não foi completamente falhado pois a preservação da *label* foi feita apesar de ter preservado uma *label* que não é especificamente correta, apenas a sua categoria.

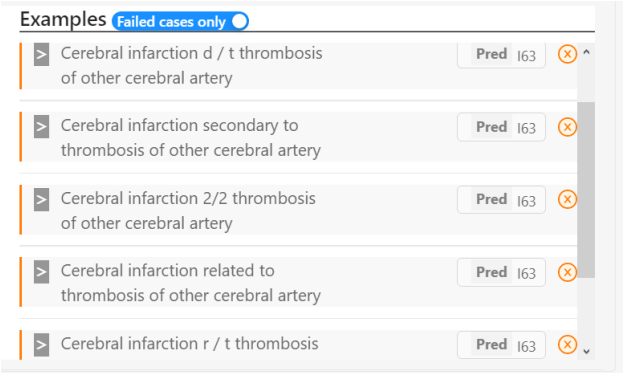


Figura 45: Resultado do terceiro teste de taxonomia no modelo de Clinical ICD-10.

5.3.2.2 *Named Entity Recognition*

Sendo este o teste principal, era necessário fazer uma grande quantidade de testes. Para isso, foram selecionados, primeiro 150 frases da nomenclatura, e depois no segundo teste, cerca de 200 frases de forma aleatória e foi aplicado o modelo, a cada uma destas. Os resultados mostram que este modelo apresenta uma percentagem de erro à volta dos 30% (Figuras 46 e 47). Apesar desta percentagem podemos concluir que o modelo é mais preciso do que o valor representa, isto porque muitas das frases em que o resultado foi errado, o modelo acertava na categoria correta apenas errando a especificação do código, o último digito. Um exemplo disso são os casos presentes na figura 47) em que o modelo deteta o código A5005 quando deveria ser o código A501, sendo que a porção A50 representa todos os tipos de

Congenital Syphilis. Além disso é importante ter em conta que o modelo está designado para atuar em contexto de nota clínica.

Lista de Tabelas 5.5: Processo de perturbação das frases originais e adição do teste INV - Typos

```
t = Perturb.perturb(phrases, Perturb.add_typos, keep_original=False)
test_inv_typos = INV(**t,
                 labels=labels_examples,
                name = 'INV - Typos',
                capability = 'Robustness')
```

```
Vocabulary - MFT 0

MFT 0
Test cases:      150
Fails (rate):    43 (28.7%)

Example fails:
A5049 Other late congenital syphilis, symptomatic
----
D860 Sarcoidosis of lung with sarcoidosis of lymph nodes
----
no_code_det Anal fissure, unspecified
----
```

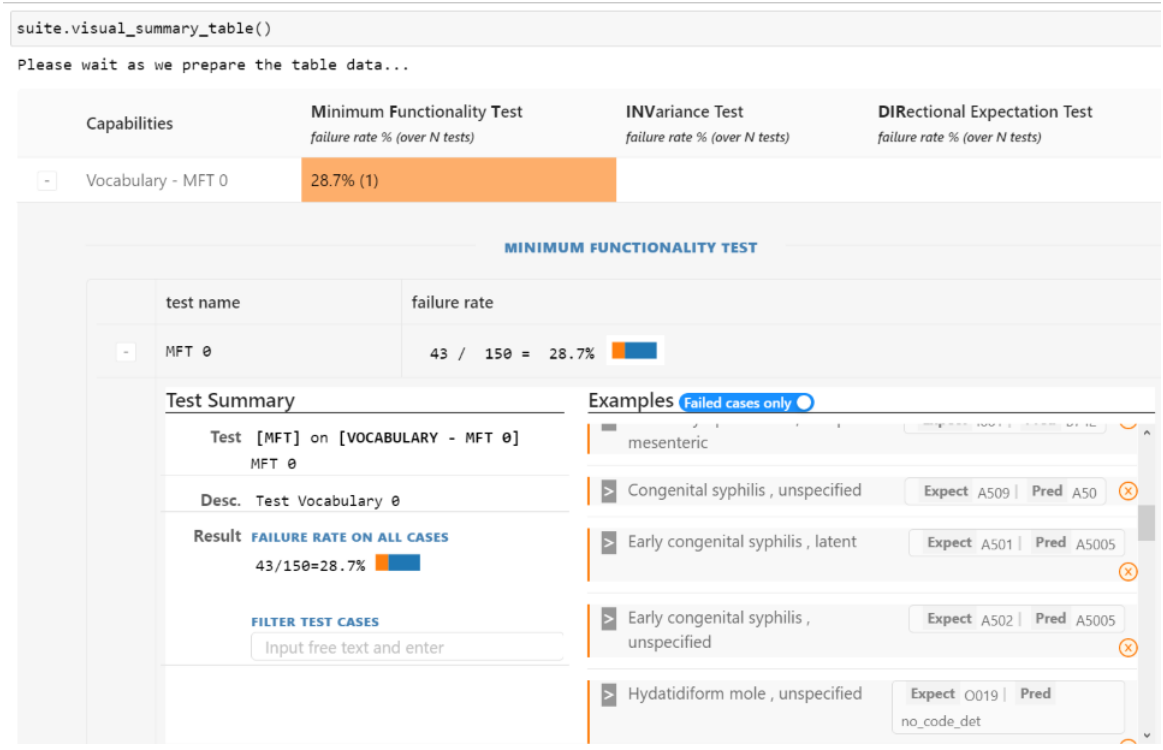


Figura 46: Resultado do teste com 150 frases para o modelo de resolução ICD-10.

```
Vocabulary - MFT

MFT
Test cases:      210
Fails (rate):    77 (36.7%)
```

Figura 47: Resultado do teste com 210 frases para o modelo de resolução ICD-10.

5.4 Discussão

A ferramenta utilizada para estes testes é, de facto, uma boa ferramenta e que apresenta uma metodologia bastante interessante e útil para a comunidade de PLN. Apesar de tudo, por ser relativamente recente, apresenta ainda algumas incoerências principalmente quando os modelos a testar não são *question-answering* e o suporte para a mesma é praticamente nulo. Não tendo sido possível obter resposta do autor da ferramenta em tempo útil para entrega da dissertação, não foram possíveis realizar os testes de *Robustness* em ambos os modelos.

Acerca dos testes realizados, os resultados são bastantes satisfatórios. Ambos os modelos cumprem relativamente bem a tarefa para a qual foram desenvolvidos e apresentam bom comportamento em relação às alterações efetuadas nas frases. Além disso, tendo em conta que os modelos são treinados em notas clínicas que são constituídas por várias frases, os resultados mostram que o modelo consegue lidar bem também com frases simples.

Ainda acerca desta secção fica o possível trabalho futuro de continuar a implementar esta metodologia de testes nos modelos e o acompanhamento do desenvolvimento da ferramenta.

Capítulo 6

Conclusões e Trabalho Futuro

Este capítulo descreve a síntese do trabalho realizado nesta dissertação bem como as perspetivas de trabalho futuro da mesma.

6.1 Conclusão

A anotação automática de informação clínica e, no geral, a aplicação de técnicas e modelos de PLN apresentam alguns desafios, sendo os principais a preservação da privacidade do paciente e o problema da estruturação dos registos clínicos que serão anotados. A formatação e o estilo dos registos clínicos variam muitas vezes, dependendo da instituição de onde são originários. A não existência de uma estrutura geral para os registos médicos eletrónicos constitui um problema para os modelos de PLN.

A falta de dados de acesso livre constitui outro dos problemas para a comunidade de PLN. A utilização de técnicas de aumento de quantidade de dados ou a anotação manual têm sido algumas das tentativas de soluções sem grandes resultados. Desde a implementação dos registos clínicos eletrónicos em hospitais, o *copy-paste* passou a ser um técnica utilizada diariamente por profissionais de saúde levando a que seja gerada muita informação redundante entre esses registos.

No decorrer desta dissertação, foi descrito o trabalho realizado e o projeto desenvolvido, começando pelo estado atual do tema da dissertação até ao desenvolvimento de uma aplicação *web* para anotação automática de informação clínica direcionada para os profissionais de saúde.

No capítulo 2 são atingidos os pimeiros objetivos propostos na introdução desta dissertação. Assim, é feita uma pesquisa extensiva e são estudadas as abordagens existentes no âmbito do tema desta dissertação. Na parte final deste capítulo é também feita a análise das ferramentas já existentes com maior relevância no mercado e, ao mesmo tempo, uma recolha das suas limitações que permitiu desenvolver

uma aplicação mais completa e que resolvesse estes problemas.

O terceiro objetivo desta dissertação, o desenvolvimento da aplicação, é descrito nos capítulos 3 e 4 desde a escolha dos dados clínicos, passando pelos modelos e a arquitetura e finalizando com a implementação da aplicação. Assim, os dados clínicos utilizados nesta dissertação foram os do *dataset* MIMIC-III. Estas notas clínicas não-identificadas provenientes de um grupo clínico dos Estados Unidos da América disponível à comunidade mediante o cumprimento de certos requisitos, sendo uma das principais contribuições para a evolução da área. Relativamente à escolha para os modelos, a utilização das bibliotecas *Spark NLP for Healthcare* treinados especificamente em notas clínicas semelhantes às presentes nessa base de dados diminuiu, bastante, a problemática relacionada à estruturação das notas clínicas. Assim, os 2 maiores obstáculos à anotação automática de informação clínica foram ultrapassados. Por fim, no capítulo 4 é detalhado todo o processo de implementação dos modelos e da aplicação em si, estando descrita toda a organização da mesma bem como as suas ligações e o seu funcionamento.

Os objetivos 4 e 5 desta dissertação são atingidos no capítulo 5. Primeiramente é feita uma análise SWOT da aplicação que permite estudar e avaliar a aplicação no mercado onde esta se insere. Na secção seguinte através de uma ferramenta de teste bastante recente, a *CheckList*, são testados os modelos e avaliada a precisão e qualidade dos mesmos.

Esta dissertação, como enunciado na motivação, pretendia providenciar uma ferramenta aos clínicos que justificasse a constante informatização da Saúde e aproveitasse o enorme desenvolvimento na área do PLN. Esta aplicação pretendia auxiliar os profissionais na análise de historiais e relatórios dos seus pacientes, poupando-lhes trabalho e reduzindo a quantidade de tempo a analisar informação redundante, repetida ou inútil.

Desta forma, com a realização de todos os objetivos, é possível anotar automaticamente informação clínica e auxiliar o trabalho dos profissionais de saúde, proporcionando-lhes, assim, menos carga de trabalho desnecessária e, consequentemente, melhores condições físicas e psicológicas para as demais tarefas adjacentes à sua profissão.

6.2 Trabalho Futuro

Como trabalho futuro pretende-se, primeiramente, a melhoria contínua da aplicação em conjunto com os profissionais de saúde, com o objetivo de perceber melhor quais as necessidades destes e desenvolver a aplicação nesse sentido de suavizar a carga profissional dos clínicos.

Além da aplicação, o trabalho junto das bibliotecas *Spark NLP* oferece inúmeras possibilidades de melhorias pois estas encontram-se diariamente em expansão e atualização, tornando-se, a cada dia, mais completas e com maior precisão. Juntamente com a aplicação e melhoria dos modelos, os testes podem também ser escalados e completados no futuro para tornar os resultados mais credíveis e analisar os modelos que sejam lançados para manter a aplicação o mais precisa possível, no presente e no futuro.

Um dos pontos de mais importância para trabalho futuro seria a criação de uma ferramenta destas para dados portugueses. Primeiramente seria fundamental o acesso e a requisição de notas clínicas portuguesas junto das instituições competentes. Esta ferramenta poderia trazer grandes vantagens ao desenvolvimento da área da Saúde em Portugal. Além disso poder colocar esta ferramenta em ambientes reais, dar-nos-ia uma perspetiva real do impacto da mesma e, de como pode ser realmente vantajoso o seu uso no dia-a-dia de um médico.

Bibliografia

[1] T. Oliveira, F. Gonçalves, P. Novais, K. Satoh e J. Neves. "OWL-based acquisition and editing of computer-interpretable guidelines with the CompGuide editor". Em: *Expert Systems* 36 (3 jun. de 2019), e12276. issn: 1468-0394. doi: 10.1111/EXSY.12276. url: https://onlinelibrary.wiley.com/doi/full/10.1111/exsy.12276https://onlinelibrary.wiley.com/doi/abs/10.1111/exsy.12276https://onlinelibrary.wiley.com/doi/10.1111/exsy.12276.

[2] J. Ramos, T. Oliveira, K. Satoh, J. Neves e P. Novais. "Cognitive Assistants—An Analysis and Future Trends Based on Speculative Default Reasoning". Em: *Applied Sciences 2018, Vol. 8, Page 742* 8 (5 mai. de 2018), p. 742. issn: 20763417. doi: 10.3390/APP8050742. url: https://www.mdpi.com/2076-3417/8/5/742/htmhttps://www.mdpi.com/2076-3417/8/5/742.

[3] "A dynamic default revision mechanism for speculative computation". Em: *Autonomous Agents and Multi-Agent Systems* 31 (3 mai. de 2017), pp. 656–695. issn: 15737454. doi: 10.1007/S10458-016-9341-9/TABLES/1. url: https://link.springer.com/article/10.1007/s10458-016-9341-9.

[4] T. Oliveira, A. Silva, J. Neves e P. Novais. "Decision Support Provided by a Temporally Oriented Health Care Assistant". Em: *Journal of Medical Systems 2016 41:1* 41 (1 nov. de 2016), pp. 1–13. issn: 1573-689X. doi: 10.1007/S10916-016-0655-6. url: https://link.springer.com/article/10.1007/s10916-016-0655-6.

[5] A. Escoval e A. C. Fernandes. *PLANO NACIONAL DE SAÚDE 2011-2016*. Rel. téc. 2011.

[6] *Registo de Saúde Eletrónico – SPMS*. url: https://www.spms.min-saude.pt/2020/07/registo-de-saude-eletronico/.

[7] T. Oliveira, P. Novais e J. Neves. "Development and implementation of clinical guidelines: An artificial intelligence perspective". Em: *Artificial Intelligence Review 2013 42:4* 42 (4 mar. de 2013), pp. 999–1027. issn: 1573-7462. doi: 10.1007/S10462-013-9402-2. url: https://link.springer.com/article/10.1007/s10462-013-9402-2.

[8] T. A. Koleck, C. Dreisbach, P. E. Bourne e S. Bakken. *Natural language processing of symptoms documented in free-text narratives of electronic health records: A systematic review*. 2019. doi: 10.1093/jamia/ocy173.

[9] P. Novais, T. Oliveira e J. Neves. "Moving towards a new paradigm of creation, dissemination, and application of computer-interpretable medical knowledge". Em: *Progress in Artificial Intelligence 2016 5:2* 5 (2 fev. de 2016), pp. 77–83. issn: 2192-6360. doi: 10.1007/S13748-016-0084-2. url: https://link.springer.com/article/10.1007/s13748-016-0084-2.

[10] C. Friedman e N. Elhadad. "Natural language processing in health care and biomedicine". Em: *Biomedical Informatics: Computer Applications in Health Care and Biomedicine: Fourth Edition* (jan. de 2014), pp. 255–284. doi: 10.1007/978-1-4471-4474-8_8.

[11] F. d. M. da Universidade do Porto. *Registos Clínicos Electrónicos*. url: http://im.med.up.pt/epr/epr.html.

[12] *Informática Médica*. url: http://im.med.up.pt/epr/epr.html.

[13] M. F. Rodrigues Leal. *Avaliação da Qualidade do Registo Clínico Eletrónico*. 2013. url: https://repositorium.sdum.uminho.pt/bitstream/1822/27778/1/dissertacao{_}MariaFilipaRodriguesLeal{_}2013.pdf.

[14] *How Portugal is advancing the use of eHealth in Europe | Healthcare IT News*. url: https://www.healthcareitnews.com/news/emea/how-portugal-advancing-use-ehealth-europe.

[15] *SNOMED - Home | SNOMED International*. url: https://www.snomed.org/.

[16] *SPMS – Serviços Partilhados do Ministério da Saúde, EPE – SNS*. url: https://www.sns.gov.pt/entidades-de-saude/servicos-partilhados-do-ministerio-da-saude/.

[17] *Semântica Categoria - Centro de Terminologias Clínicas*. url: https://www.ctc.min-saude.pt/category/terminologias/interoperabilidade-semantica/.

[18] J. C. Feblowitz, A. Wright, H. Singh, L. Samal e D. F. Sittig. "Summarization of clinical information: A conceptual model". Em: *Journal of Biomedical Informatics* (2011). issn: 15320464. doi: 10.1016/j.jbi.2011.03.008.

[19] S. Berndorfer e A. Henriksson. "Automated Diagnosis Coding with Combined Text Representations". Em: *Studies in Health Technology and Informatics* (2017). issn: 18798365. doi: 10.3233/978-1-61499-753-5-201.

[20] *Tf-idf :: A Single-Page Tutorial - Information Retrieval and Text Mining*. url: http://www.tfidf.com/.

[21] *Implementing Deep Learning Methods and Feature Engineering for Text Data: The Continuous Bag of Words (CBOW)*. url: https://www.kdnuggets.com/2018/04/implementing-deep-learning-methods-feature-engineering-text-data-cbow.html.

[22] T. Mikolov, K. Chen, G. Corrado e J. Dean. "Efficient estimation of word representations in vector space". Em: *1st International Conference on Learning Representations, ICLR 2013 - Workshop Track Proceedings*. 2013. arXiv: 1301.3781.

[23] "Creating an automated trigger for sepsis clinical decision support at emergency department triage using machine learning". Em: *PLoS ONE* (2017). issn: 19326203. doi: 10.1371/journal.pone.0174708.

[24] *What is sepsis? | Sepsis | CDC*. url: https://www.cdc.gov/sepsis/what-is-sepsis.html.

[25] W. W. Chapman, W. Bridewell, P. Hanbury, G. F. Cooper e B. G. Buchanan. "A simple algorithm for identifying negated findings and diseases in discharge summaries". Em: *Journal of Biomedical Informatics* 34.5 (2001), pp. 301–310. issn: 15320464. doi: 10.1006/jbin.2001.1029.

[26] *MALLET homepage*. url: http://mallet.cs.umass.edu/.

[27] I. Spasic e G. Nenadic. *Clinical text data in machine learning: Systematic review*. 2020. doi: 10.2196/17984.

[28] Y. H. Su, C. P. Chao, L. C. Hung, S. F. Sung e P. J. Lee. "A natural language processing approach to automated highlighting of new information in clinical notes". Em: *Applied Sciences (Switzerland)* 10.8 (2020). issn: 20763417. doi: 10.3390/APP10082824.

[29] *Text Annotation Tools for NLP. The goal of this article is to give a... | by Javier Ramos | ITNEXT*. url: https://itnext.io/text-annotation-tools-for-nlp-c254f8ee52f7.

[30] *About MIMIC | MIMIC*. url: https://mimic.mit.edu/docs/about/.

[31] "MIMIC-III, a freely accessible critical care database". Em: *Scientific Data 2016 3:1* 3 (1 mai. de 2016), pp. 1–9. issn: 2052-4463. doi: 10.1038/sdata.2016.35. url: https://www.nature.com/articles/sdata201635.

[32] J. P. Monteiro, D. Ramos, D. Carneiro, F. Duarte, J. M. Fernandes e P. Novais. "Meta-learning and the new challenges of machine learning". Em: *International Journal of Intelligent Systems* 36 (11 nov. de 2021), pp. 6240–6272. issn: 1098-111X. doi: 10.1002/INT.22549. url: https://onlinelibrary.wiley.com/doi/full/10.1002/int.22549https://onlinelibrary.wiley.com/doi/abs/10.1002/int.22549https://onlinelibrary.wiley.com/doi/10.1002/int.22549.

[33] J. Uszkoreit. "Transformer : A Novel Neural Network Architecture for Language Understanding". Em: *Google AI Blog* (2017).

[34] K. S. Nugroho, A. Y. Sukmadewa e N. Yudistira. "Large-Scale News Classification using BERT Language Model: Spark NLP Approach". Em: (jul. de 2021). url: https://arxiv.org/abs/2107.06785v2.

[35] J. Devlin, M.-W. Chang, K. Lee e K. Toutanova. "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding". Em: *NAACL HLT 2019 - 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies - Proceedings of the Conference* 1 (out. de 2018), pp. 4171–4186. url: https://arxiv.org/abs/1810.04805v2.

[36] O. Kovaleva, A. Romanov, A. Rogers e A. Rumshisky. "Revealing the dark secrets of Bert". Em: 2020. doi: 10.18653/v1/d19-1445.

[37] "A primer in bertology: What we know about how bert works". Em: *Transactions of the Association for Computational Linguistics* 8 (2020). issn: 2307387X. doi: 10.1162/tacl_a_00349.

[38] J. S. Lee e J. Hsiang. "Patent classification by fine-tuning BERT language model". Em: *World Patent Information* 61 (jun. de 2020), p. 101965. issn: 0172-2190. doi: 10.1016/J.WPI.2020.101965.

[39] A. Adhikari, A. Ram, R. Tang, W. L. Hamilton e J. Lin. "Exploring the Limits of Simple Learners in Knowledge Distillation for Document Classification with DocBERT". Em: 2020. doi: 10.18653/v1/2020.repl4nlp-1.10.

[40] J. Lee, W. Yoon, S. Kim, D. Kim, S. Kim, C. H. So e J. Kang. "BioBERT: A pre-trained biomedical language representation model for biomedical text mining". Em: *Bioinformatics* 36 (4 2020). issn: 14602059. doi: 10.1093/bioinformatics/btz682.

[41] I. Beltagy, K. Lo e A. Cohan. "SCIBERT: A pretrained language model for scientific text". Em: 2020. doi: 10.18653/v1/d19-1371.

[42] X. P. Qiu, T. X. Sun, Y. G. Xu, Y. F. Shao, N. Dai e X. J. Huang. *Pre-trained models for natural language processing: A survey*. 2020. doi: 10.1007/s11431-020-1647-3.

[43] "Pre-Trained Models: Past, Present and Future". Em: *AI Open* (2021). issn: 26666510. doi: 10.1016/j.aiopen.2021.08.002.

[44] V. Kocaman e D. Talby. "Spark NLP: Natural Language Understanding at Scale". Em: *Software Impacts* 8 (mai. de 2021), p. 100058. issn: 2665-9638. doi: 10.1016/J.SIMPA.2021.100058.

[45] P. Goyal, S. Pandey e K. Jain. *SpaCy*. 2018.

[46] *2020 NLP Survey Report - Gradient Flow*. url: https://gradientflow.com/2020nlpsurvey/.

[47] C. Ambika. *5 Reasons Why Spark NLP Is The Most Widely Used Library In Enterprises*. url: https://analyticsindiamag.com/5-reasons-why-spark-nlp-is-the-most-widely-used-library-in-enterprises/.

[48] M. A. Kaya. *SpaCy or Spark NLP — A Benchmarking Comparison*. Ago. de 2020. url: https://medium.com/spark-nlp/spacy-or-spark-nlp-a-benchmarking-comparison-23202f12a65c.

[49] D. Ghimire. "Comparative study on Python web frameworks: Flask and Django". Em: (2020). url: http://www.theseus.fi/handle/10024/339796.

[50] K. Larkins. "A COMPARISON OF PYTHON WEB DEVELOPMENT MICROFRAMEWORKS". Em: (2020).

[51] "Efficient Way Of Web Development Using Python And Flask". Em: *International Journal of Advanced Research in Computer Science* 6 (2). url: www.ijarcs.info.

[52] *Essential SQLAlchemy: Mapping Python to Databases - Jason Myers, Rick Copeland - Google Livros*.

[53] "Jinja2 Documentation Release 2.0 Armin Ronacher". Em: (2008).

[54] R. G. Dyson. "Strategic development and SWOT analysis at the University of Warwick". Em: *European Journal of Operational Research* 152 (3 2004). issn: 03772217. doi: 10.1016/S0377-2217(03)00062-6.

[55] P. Rajpurkar, R. Jia e P. Liang. "Know What You Don't Know: Unanswerable Questions for SQuAD". Em: (2018), pp. 784–789.

[56] "Errudite: Scalable, reproducible, and testable error analysis". Em: 2020. doi: 10.18653/v1/p19-1073.

[57] M. T. Ribeiro, T. Wu, C. Guestrin e S. Singh. "Beyond Accuracy: Behavioral Testing of NLP models with CheckList". Em: (mai. de 2020). url: https://arxiv.org/abs/2005.04118v1.