Trabalho de Conclusão de Curso

RECONHECIMENTO DE ENTIDADES CARACTERÍSTICAS PARA CLASSIFICAÇÃO DE NÓDULOS PULMONARES EM LAUDOS MÉDICOS

Lucas Agra de Omena
lao2@ic.ufal.br

Orientador:
Prof. Dr. Marcelo Costa Oliveira

Maceió, Outubro de 2023

Lucas Agra de Omena

RECONHECIMENTO DE ENTIDADES CARACTERÍSTICAS PARA CLASSIFICAÇÃO DE NÓDULOS PULMONARES EM LAUDOS MÉDICOS

Monografia apresentada como requisito parcial para obtenção do grau de Bacharel em Ciência da Computação do Instituto de Computação da Universidade Federal de Alagoas.

Orientador:

Prof. Dr. Marcelo Costa Oliveira

Maceió, Outubro de 2023

Catalogação na Fonte
Universidade Federal de Alagoas
Biblioteca Central
Divisão de Tratamento Técnico

Bibliotecário: Marcelino de Carvalho Freitas Neto – CRB-4 - 1767

O48a Omena, Lucas Agra de.

Reconhecimento de entidades características para classificação de nódulos pulmonares em laudos médicos / Lucas Agra de Omena. – 2023.

27 f. : il.

Orientador: Marcelo Costa Oliveira.

Monografia (Trabalho de conclusão de curso em Ciência da Computação) – Universidade Federal de Alagoas, Instituto de Computação. Maceió, 2023.

Bibliografia: f. 24-27.

1. Medicina. 2. Lung RADs. 3. Processamento de linguagem natural. 4. Aprendizagem de máquina. 5. Aprendizagem profunda. 6. Prova pericial. 7. Transformers. 8. Representação de Codificador Bidirecional para Transformadores. 9. SpaCy (Framework). I. Título.

CDU: 004.81:159.953.5

AGRADECIMENTOS

Gostaria de agradecer ao Instituto de Computação da UFAL e seus professores e funcionários pelo curso de alta qualidade oferecido gratuitamente que me abriu inúmeras oportunidades, possibilitando-me alcançar um sonho profissional. Agradeço especialmente ao professor Dr. Marcelo Costa que orientou meu projeto de iniciação cientifica, abrindo as primeiras portas no inicio do curso e da minha carreira, e este trabalho, em sua finalização.

Agradeço aos bons amigos que fiz durante o curso e que fizeram parte da trajetória de estudos, provas, projetos e competições, tornando isso tudo uma experiência melhor.

Agradeço aos meus amigos e colegas de trabalho pelo incentivo, demonstrando que é possível conciliar trabalho e estudo.

Agradeço a minha noiva, por ter estado ao meu lado em todos os momentos de incertezas durante esses anos de graduação.

Agradeço a Deus por ter o privilégio de poder estudar com total dedicação sem precisar me preocupar com questões básicas de alimentação, moradia e saúde. O que deveria ser um direito universal mas pouquíssimas pessoas tem acesso.

E mais importante, por ultimo, agradeço aos meus pais, que me apoiaram incondicionalmente quando eu decidi trocar de carreira e fazer uma segunda graduação. À minha mãe, meu maior exemplo de dedicação e esforço, mesmo em meio às adversidades; E ao meu pai, que esteve sempre me incentivando e sendo o meu maior torcedor: Devo tudo a vocês, muito obrigado.

RESUMO

O câncer de pulmão é o tipo de doença neoplástica maligna mais mortífera, ocasionando 1.8 milhão de vítimas em 2020, em número absoluto sendo superior a soma do segundo e terceiro colocados (câncer colorretal e de fígado). Estudos mostram que o diagnóstico precoce da doença é fundamental para aumentar as chances de sucesso do tratamento, feito majoritariamente pela detecção de nódulos pulmonares através de tomografias computadorizadas (TC) de tórax. Os resultados desses exames são tipicamente armazenados de forma não estruturada, em formato de texto livre, depois de passar por um processo que contribui para a sua corrupção, como erros gramaticais, erros de digitação, ou até mesmo falta de convenção dos termos clínicos. Essa falta de estruturação dos dados cria um obstáculo desnecessário para o diagnóstico da doença, ao ser necessário buscar nesses resultados a presença de nódulos com características malignas. Com o objetivo de identificar e extrair as informações relevantes para o diagnóstico, foi treinado um modelo utilizando técnicas de *deep-learning* para fazer o reconhecimento de entidades, permitindo a identificação, em laudos textuais de TCs de tórax, dessas entidades correspondentes às características nodulares. O método de definição das entidades in-loco levou em consideração que as entidades identificadas pelo modelo possibilitassem a classificação nodular de acordo com a diretriz internacional Lung-Rads. Definiu-se então seis entidades com o propósito de identificar *quantidade*, *tipo*, *tamanho*, *local* e *características gerais* dos nódulos além da *presença de enfisema pulmonar*. Para o treinamento do modelo foi utilizado a ferramenta spaCy, usando como base uma arquitetura de modelo de linguagem pré-treinado BERT, em português brasileiro, em que foi obtido uma medida F1 de 91,89%, utilizando um conjunto de dados com 600 documentos, cedidos pelo hospital Unimed Maceió e anotados manualmente como descrito na metodologia.

Palavras-chave: medicina; lung rads; PLN; aprendizagem de máquina; aprendizagem profunda; REN; laudos médicos; transformers; bert; spacy; português.

ABSTRACT

Lung cancer is the deadliest type of malignant neoplastic disease, causing 1.8 million victims in 2020. In absolute numbers, it is higher than the second and third leading types combined (colorectal and liver cancer). Studies show that early diagnosis is crucial to increase the chances of successful treatment, primarily achieved through the detection of lung nodules via lung computerized tomography (CT) scans. As a typical practice, the results of these scans are data stored in an unstructured, free text format, after passing through a storing process that contributes to its corruption, such as grammatical errors, typos, or even lack of convention of clinical terms. This lack of data structure creates an unnecessary obstacle for the disease diagnosis, as it is necessary to search in these results for the presence of nodules with malignant characteristics. Aiming for identifying and extracting relevant data for the disease diagnosis, a deep-learning model was trained to perform a named entity recognition task, enabling the retrieval of entities related to the nodules aspects from textual lung CT scan reports. The entities were defined taking in consideration the characteristics needed to enable the classification according to the international Lung-Rads guideline. Six entities were defined to identify the *quantity*, *type*, *size*, *location*, and *general characteristics* of the nodules, as well as the presence of *pulmonary emphysema*. The spaCy framework was used to train the model, employing a BERT pre-trained language model architecture in Brazilian Portuguese. The model achieved a F1 score of 91.89% using a dataset of 600 documents provided by Unimed Maceio Hospital, manually annotated as described in the methodology.

Key-words: healthcare; lung rads; nlp; machine learning; deep learning; NER; medical reports; transformers; bert; spacy; portuguese.

SUMÁRIO

1

Introdução

Durante os últimos anos, o avanço da inteligencia artificial (IA) vem superando as expectativas do que era previsto, isso deve-se principalmente à crescente quantidade de dados disponíveis, devido à normalização de sistemas computacionais em todos os âmbitos do cotidiano [7, 1]. Agregado ao avanço dos componentes eletrônicos, a abundancia de dados somados ao novo poder de processamento acessível tornou fértil o avanço das áreas de aprendizagem de máquina (do inglês, *machine learning* - ML) e aprendizagem profunda (do inglês, e *deep learning* - DL), utilizando desses dados para treinar modelos que reproduzem a informação ali contida, gerando valor, e por sua vez, interesse.

Com isso, a aplicação da inteligência artificial, ou mais especificamente: *machine learning*, como tem sido associada ultimamente, vem revolucionando setores de diferentes segmentos, dentre eles a medicina, por ser um dos que mais gera dados eletrônicos e com projeções ainda maiores para o futuro [10]. Esteva et al. [7] mostra que sistemas de DL já conseguem ter a mesma acurácia de médicos ao dar diagnósticos em diversas áreas da medicina. Porém, existe um consenso de que a substituição desses profissionais por máquinas seja improvável, ao menos em um futuro próximo. Em vez disso, espera-se que a tecnologia evolua para apoiar o especialista, atuando como uma ferramenta de segunda opinião e de auxílio computadorizado ao diagnostico (CAD), logo, ampliando o que pode ser oferecido ao paciente [1].

O uso da IA na medicina, no entanto, não é uma tarefa simples. O manuseio dos dados envolvem complicações que vão desde o armazenamento, disponibilidade, qualidade, privacidade, até questões éticas e legais de como eles podem ser utilizados. Para que isso seja possível, é preferível que haja um trabalho em conjunto de profissionais da tecnologia da informação e da saúde, em que ambos precisam estender suas áreas de conhecimento para além de sua especialidade, como já é visto em alguns países, onde conteúdos relacionados à tecnologia e inteligência artificial são abordados nos currículos acadêmicos dos cursos de saúde [1].

Dentre as vantagens trazidas pelo uso da IA, uma delas é a automação, integral ou parcial, de tarefas repetitivas, abrindo muitas oportunidades para seu uso na medicina, visto que há

evidencias de diversas tarefas deste tipo no trabalho tipico de um médico [10]. Além disso, assim como Ahuja [1] aponta, a IA na medicina pode ser utilizada como uma camada extra de decisão de forma a mitigar erros e simplificar processos, causando uma redução de tempo em que médicos passam revisando documentos em até 44% [10], e consequentemente permitindo que eles invistam mais tempo dedicado aos pacientes.

1.1 Contextualização do problema

O câncer está entre as principais causas de morte em todos os países do mundo, e o câncer de pulmão é o que lidera a mortalidade, ocasionando 1.8 milhão de mortes em 2020, sendo o segundo tipo mais diagnosticado, ficando atrás apenas do câncer de mama [21, 26]. Estudos mostram [4, 26] que a detecção precoce do câncer de pulmão é de vital importância para o paciente, em que a sobrevivência após 1 ano de diagnóstico pode chegar a 85% em casos de detecção no estágio I (tumor localizado), contra 15% no estágio IV (metastático). A principal característica observada para detectar o câncer de pulmão é a presença do nódulo pulmonar, definido por Silva et al. [18] como uma opacidade focal arredondada, ou pelo menos parcialmente delimitada, menor que 3,0 cm de diâmetro. Com o uso da tomografia computadorizada (TC), é possível identificá-los e observar suas características.

Na prática comum, os resultados dos exames de TC são armazenados de forma não estruturada. Normalmente, o médico responsável usa um gravador de voz para ditar o que ele observa nas imagens do exame, que por sua vez é transcrito em formato de texto por um digitador. Nesse formato, os dados armazenados ficam desorganizados, sem um padrão determinado, com erros de digitação e erros gramaticais, o que dificulta a sua interpretação e impossibilita seu uso em sistemas digitais.

O uso de ML em combinação com o processamento de linguagem natural (do inglês, *Natural Language Processing* - NLP) têm sido aplicado e dado utilidade aos dados não estruturados [17]. A tarefa de reconhecimento de entidades nomeadas (do inglês, *Named Entity Recognition* - NER) pode ser utilizada para identificar e extrair entidades de interesse em um texto qualquer. Por exemplo, supondo um laudo de TC em formato de texto, é possível utilizar o NER para identificar entidades relacionadas às características nodulares, extraindo informações úteis dos dados não estruturados.

Lung-RADS [14] é uma diretriz definida pelo *American College of Radiology*, utilizada globalmente para padronizar a classificação de periculosidade de nódulos pulmonares com base em suas características. Para isso, os médicos precisam avaliar as características dos nódulos presentes nas TCs e então enquadrar em um dos níveis descritos pela tabela. Essa tarefa é particularmente sujeita a erros humanos devido às diversas combinações de características que definem as classificações, e ao mesmo tempo uma boa candidata ao uso de *machine learning*, visto que existem regras explicitas definidas para a classificação, o que possibilita a automati-

zação do processo.

1.2 Objetivo do estudo

O estudo tem como principal objetivo organizar os dados não-estruturados presentes nos laudos de TC de tórax transcritos, utilizando NLP para identificar e estruturar as informações clínicas de importância. Levando em consideração os critérios de classificação da diretriz Lung-RADS, estima-se treinar um modelo para realizar a tarefa de NER à ponto de identificar as principais características nodulares, utilizando técnicas de DL através da ferramenta de código aberto chamada SpaCy [9]. Dessa forma, diminuindo o atrito necessário para fazer a classificação do grau de malignidade nodular de acordo com a diretriz, e viabilizando o seu uso por sistemas digitais.

O objetivo secundário é analisar a performance do modelo treinado com diferentes quantidades de dados, investigando a viabilidade de aplicação dessa metodologia em diferentes casos de uso.

2

Revisão da literatura

No ano de 2017, cientistas da Google lançaram o artigo chamado *Attention Is All You Need* [22], apresentando um novo tipo de arquitetura para processamento de texto, cujo foco seria a tradução textual entre idiomas. O artigo apresentou um novo tipo modelo chamado de *Transformers* que usa como fundamento a relação entre cada par de palavras (*tokens*) de um determinado texto, através de um mecanismo chamado de *Camada de Atenção*. O mecanismo possibilita definir o grau de importância que uma palavra tem com outra, mesmo que não estejam posicionalmente próximas, o que mudou a forma de como os modelos interpretam semanticamente o texto, atingindo o estado da arte na tarefa de tradução Inglês-Francês, utilizando uma fração do custo de treinamento dos melhores modelos disponíveis na época.

Novamente, em 2019, outro estudo da Google [6] apresenta um novo modelo de linguagem baseado na arquitetura *Transformers*, chamado de **BERT** *(**B**idirectional **E**ncoder **R**epresentation from **T**ransformers)*, que utiliza *Camadas de Atenção* bidirecionais para gerar um modelo de linguagem treinado de forma não-supervisionada. Isto é, os dados de treinamento (*corpus*) não precisam ser anotados por um especialista, permitindo seu treinamento com gigantescos *corpus* extraídos da internet.

O objetivo de treinamento para o modelo BERT é: dado uma frase com uma palavra mascarada, adivinhar tal palavra. Dessa forma são compostos os *corpus* de treinamento do BERT: milhões de frases com uma palavra mascarada cada, onde a função-objetivo é adivinhar tal palavra baseando-se no contexto da frase e sua posição. Tal treinamento acaba criando um modelo de linguagem capaz de discernir contexto e semântica entre frases e palavras.

O modelo BERT requer um custo altíssimo para ser treinado, porém, a virada de jogo acontece ao utilizar a técnica de transferência de aprendizagem [2, 28] (*transfer learning* ou *fine-tuning*) que permite reutilizar a informação aprendida em um modelo já treinado para realizar outra tarefa. Essencialmente, tem-se um grande modelo treinado que contem toda informação de uma linguagem condensada (chamado de modelo de linguagem pré-treinado), que é submetido a um pequeno treino – que pode ser visto como um ajuste – para direcioná-lo a realizar

uma outra tarefa, reutilizando o que ja havia sido aprendido antes. O processo é viável mesmo ao utilizar um *corpus* pequeno e um poder de processamento limitado. Dessa forma, o artigo apresentado atingiu o estado da arte em **onze** diferentes tarefas de PLN.

2.1 Visão geral de IA e suas aplicações na medicina

Os sistemas clássicos de *machine learning* conseguem derivar regras utilizando estatística e características pré-definidas dos dados recebidos (*features*). Sistemas de *deep-learning*, um subtipo de *machine learning*, por sua vez, conseguem abstrair a tarefa de definição das *features*, superando os sistemas clássicos quando há grandes quantidades de dados, e tendem a ficar cada vez melhores à medida que essa quantidade aumenta. Isso acontece pois os modelos de *deep-learning* possuem mais camadas, permitindo que consigam extrair padrões mais complexos dos dados, desde que haja uma quantidade suficiente.

Estudos evidenciam que devido à crescente quantidade de dados gerados na área da saúde e ao aumento do uso de dispositivos eletrônicos e serviços digitais, os avanços em *deep-learning* têm um potencial de trazer benefícios significativos a este campo [7, 10, 1]. Um dos potenciais ganhos são avanços na classificação de imagens e detecção de objetos, mostrando resultados promissores de diagnósticos nas áreas de radiologia, oftalmologia, dermatologia, onde sistemas inteligentes podem revisar ou apenas sugerir resultados. Dentre os avanços, já existem relatos de modelos de deep-learning na área de visão computacional que demonstram ter uma precisão comparável à de médicos especializados [7]. Estima-se que ainda nesta década imagens médicas sejam pré-analisadas por um sistema inteligente antes de chegar ao médico, de acordo com Ahuja [1].

Na área de PLN, evidenciam-se soluções que focam em economizar tempo de trabalho médico, como sintetização e classificação de texto, assistentes virtuais, anotação de imagens e extração de informação, como forma de diminuir o tempo que médicos gastam trabalhando com documentação de dados clínicos, que pode corresponder à metade de sua jornada de trabalho, impossibilitando investir mais tempo com os pacientes [7].

Aponta-se que a geração de dados não-estruturados na forma de registro digitais serão rapidamente onipresentes e crescem cada vez mais. Uma única hospitalização pode gerar em torno de 150.000 dados [7], por exemplo. O resultado de sistemas inteligentes com acesso a essa quantidade de dados ao longo do tempo poderia superar qualquer conhecimento que uma pessoa individual seria capaz de deter, dando cobertura a todo tipo de doenças e condições raras. Um dos pontos de risco, no entanto, é que os dados não sejam tratados de forma adequada e isso possa levar a erros médicos. Segundo Ahuja [1], entidades médicas que começarem a se familiarizar com o tratamento de dados de maneira profissional e responsável cedo, terão mais facilidade e naturalidade em implantar sistemas inteligentes conforme sua evolução.

Há também muitos outros casos de uso de *deep-learning*, como por exemplo no estudo

de doenças e desenvolvimento de novas drogas. Recentemente a primeira droga desenvolvida inteiramente por inteligência artificial entrou em fase de testes com humanos. De acordo com Fox News [8], a nova droga, chamada de *INS018_055* foi desenvolvida para tratar a doença *Fibrose Pulmonar Idiopática*, atualmente incurável e que geralmente leva à morte entre dois e cinco anos.

> "However, with the latest advances in artificial intelligence, it was developed much faster than traditional drugs, [...]. AI allows us to analyze massive quantities of data and find connections that human scientists might miss, and then 'imagine' entirely new molecules that can be turned into drugs,' - *diz Alex Zhavoronkov (CEO da Insilico Medicine)* [8].

2.2 Trabalhos relacionados a extração de entidades na medicina

Há muito tempo a tarefa de extração de informações de textos clínicos não-estruturados é um desafio. Antes do *deep-learning* tornar-se acessível, era comum serem utilizados métodos de *machine learning* baseados em heurísticas e dicionários para extração de palavras-chave [13]. Sugimoto et al. [20] aponta que um grande problema desses sistemas é a necessidade de identificar palavras exatas, o que impossibilita o seu uso fora do ambiente onde foi desenvolvido, em que os textos podem apresentar outros padrões, erros ortográficos, abreviações, ou até outro idioma. Também é destacado que esses tipos de sistemas falham particularmente quando outro idioma além do inglês é considerado, visto que eles dependem diretamente de ferramentas de dicionario que centralizam as terminologias clinicas, majoritariamente disponíveis apenas em inglês, por exemplo: UMLS [5], HITEx [27] e MetaMap [3].

Nos últimos anos, novas técnicas de *machine learning* surgiram superando os resultados de técnicas baseadas em dicionários, como o método apresentado por Lample et al. [11, 2016], utilizando *CRF* (***C**onditional **R**andom **F**ield*) e *BiLSTM* (***Bi**directional **L**ong **S**hort-**T**erm **M**emory*). Mais recentemente surgiram as técnicas baseadas em modelos de linguagem, como o ELMo [15, 2018] e BERT [6, 2019], superando mais uma vez o estado da arte em várias tarefas de PLN. Si et al. [16, 2019] faz uso deles no contexto de dados clínicos e constata resultados superiores em relação as arquiteturas antigas, provando que usar modelos de linguagens pré-treinados trazem resultados melhores. Diferentemente das técnicas anteriores, esses novos métodos são aplicáveis à idiomas diferentes do inglês, desde que haja um modelo de linguagem pré-treinado no idioma desejado [13].

3

Metodologia

Neste capitulo será abordado em detalhe todo o processo necessário para o treinamento de um modelo de extração de entidades baseado em *deep-learning*, utilizando a ferramenta spaCy.

Devido as vantagens citadas anteriormente, será utilizado a técnica de *fine-tuning* (um tipo de *transfer learning* [28]) em um modelo de linguagem baseado em contexto (BERT) já treinado em português brasileiro, como ponto de partida do treinamento que será feito. Isso permite que não precisemos nos preocupar em ensinar ao modelo sobre o idioma, mas somente ensina-lo a fazer a tarefa desejada: extrair entidades. Por exemplo, um modelo BERT pré-treinado em português já consegue distinguir que nas frases "Vou ao **banco** sacar dinheiro." e "O **banco** da praça está sujo." a palavra **banco** tem significados diferentes. Essa habilidade de entender o contexto das palavras contribui para que as entidades corretas sejam extraídas e falsos positivos sejam evitados [12].

Para isso, o modelo pré-treinado escolhido foi o BERTimbau-Base[19], versão *cased* (isto é, diferencia letras maiúsculas de minusculas), possuindo 12 camadas e 110 milhões de parâmetros. Foi treinado por 1.000.000 *steps* com o corpus BrWaC (Brazilian Web as Corpus) [24], que possui 145 milhões de frases e 2.7 bilhões de tokens. *Step* é o nome dado a uma iteração durante o processo de treinamento, que será melhor descrito na seção 3.5.

3.1 Descrição do framework spaCy e suas características

SpaCy [9] é um framework popular de PLN de código aberto e desenvolvido em Python. Ele oferece uma série de recursos e funcionalidades que ajudam com tarefas de processamento de texto, como tokenização, análise sintática, reconhecimento de entidades nomeadas, extração de informações, lematização, entre outros. Ele foi construído de forma a ser eficiente e intuitivo e pode ser utilizado em larga escala, em vários idiomas. O spaCy funciona em vários graus de abstração, permitindo carregar e servir um modelo já treinado rapidamente, mas ao mesmo

tempo possuindo uma arquitetura modular que permite personalizações avançadas, possibilitando o treinamento de modelos próprios, e suporte para utilização de modelos treinados por terceiros. Por exemplo, o spaCy dá suporte diretamente ao uso de modelos disponíveis no *HuggingFace [25]*, uma plataforma que funciona como uma biblioteca de modelos baseados em *Transformers*, onde hoje encontram-se a maioria dos modelos públicos mais utilizados – dentre eles o BERTimbau, utilizado neste estudo –, permitindo assim uma gama de possibilidades.

3.2 Conjunto de dados utilizado e seu tratamento

O corpus utilizado constitui-se de documentos referentes a laudos médicos que descrevem os resultados de TCs de tórax, cedidos pelo hospital Unimed de Maceió, Alagoas. O corpus, no entanto, possui apenas os laudos em formato de texto "crús", da forma como foram escritos e digitalizados, o que não é suficiente para treinar o modelo de extração de entidades. Para isso, as entidades que estão contidas nesse texto e que desejamos extrair precisam estar sinalizadas, para que sirvam de exemplos do que deve ser extraído durante o treinamento do modelo. Esse tipo de treinamento é chamado de supervisionado, pois utiliza dados classificados ou anotados por um especialista, que durante o treinamento é considerado como a resposta correta (também chamado de *gold standard*). Esse processo de marcação é chamado de *Anotação das entidades*, e será detalhado em seguida.

Antes disso, primeiro foi necessário pré-processar os documentos utilizando técnicas de expressões regulares para remover padrões indesejados e manter a homogeneidade dos dados. Foram removidas quebra de linhas, linhas em branco, tabulações e espaçamentos múltiplos indesejados, de forma a manter cada documento como um único parágrafo de texto.

É importante notar que na etapa de treinamento supervisionado será utilizado como base um modelo de linguagem treinado de forma não-supervisionada (BERTimbau). Por isso, esse processo completo é considerado um método semi-supervisionado.

3.3 Definição dos tipos de entidades

Com o objetivo de extrair de laudos médicos as entidades que sirvam para classificar os nódulos de acordo com a tabela Lung-Rads [14], foram considerados alguns aspectos para a definição dos tipos dessas entidades, dentre eles:

1. Quais as características mais importantes para a classificação nodular levadas em conta na tabela de diretriz Lung-Rads;

2. O nível de detalhamento das características dos nódulos presentes nos documentos;

3. A quantidade de documentos disponíveis e a quantidade estimada a serem anotadas, considerando a mão de obra disponível;

4. O número mínimo de tipos de entidades necessárias para satisfazer todos os critérios, de forma que sejam bem definidas, evitando interseção entre seus contextos.

Essas considerações foram feitas pois em um cenário perfeito seria possível ter diversos tipos de entidades, que conseguiriam descrever perfeitamente todas as características dos nódulos. Porém, um número elevado de entidades dificulta a tarefa que o modelo terá de aprender, já que ele precisaria considerar mais tipos de respostas e casos diferentes. Além disso, seria necessário um *corpus* muito maior, com mais documentos anotados, de forma que cobrisse várias situações diferentes com cada tipo de entidade. Também é importante definir claramente cada tipo de entidade pois durante a anotação deve-se evitar ao máximo casos em que haja incerteza, confusão ou sobreposição de contexto entre as anotações, com o objetivo de deixar claro para o modelo o que cada entidade representa e como elas se diferenciam. Entidades com escopos bem definidos têm sua identificação facilitada, melhorando sua acurácia. Foram então definidas **seis tipos de entidades**.

NoduleCount

Entidade que tenta captar a quantidade de nódulos mencionado no texto do documento. Por vezes o documento especifica exatamente uma quantidade, mas foi observado que na maioria dos casos contem termos genéricos como “múltiplos nódulos” ou “nódulos”. Também foi observado que acontece em diversas ocasiões de haver mais de uma anotação de nódulo em um mesmo documento, com características diferentes, sendo descritos em diferentes partes texto. Nesses casos a tarefa de extração de entidades não consegue responder a quantidade exata de nódulos, mas sim extrairá as entidades presentes no texto a quantidade de vezes que for mencionada.

NoduleSize

Entidade que descreve especificamente textos relacionados ao tamanho dos nódulos. Durante a marcação, foram ignorados termos genéricos como “grande” ou “pequeno”, e sim marcados termos que especificam exatamente o diâmetro, área ou volume em unidades de medida, normalmente *cm* ou *mm*. Isso foi definido para garantir uma homogeneidade nos dados extraídos, melhorar a especificidade da entidade e facilitar que o dado seja usado na classificação Lung-Rads, visto que termos genéricos não trazem tanto valor para a classificação quando trata-se do tamanho e misturá-los com valores específicos poderia dificultar o processo de classificação.

NoduleType

Foram considerados os termos que definem a principal característica de composição do nódulo e que está de acordo com a Lung-Rads. Termos como “sólido”, “não sólido”, “aéreo”, “vidro fosco”, “calcificado” e “densidade de partes moles”.

NoduleLocation

Busca representar a parte do texto que descreve a localidade anatômica do nódulo descrito. Utilizar um modelo de linguagem que leva em consideração o contexto é particularmente benéfico neste caso, pois permite que o modelo faça a distinção e reconheça a entidade apenas quando a localização descrita se referir a um nódulo e não a qualquer outro tipo de substantivo médico, também presentes nos documentos.

NoduleInfo

Durante a análise dos documentos foi percebido que além do tipo, tamanho e local dos nódulos, os médicos descrevem também outros tipos de características variadas. Como não seria viável definir diversos tipos de entidades para cada uma dessas características, devido à suas heterogeneidades e escassez, seriam muitas entidades diferentes com poucos exemplos. Definiu-se então a entidade *NoduleInfo* para agregar todos esses tipos de características adicionais que não se englobam nas características principais definidas acima. Dessa forma, os dados extraídos nessa entidade não são diretamente relacionados às classificações Lung-Rads, mas podem ser úteis como informação complementar, dependendo do método que será abordado para fazer essa classificação.

Emphysema

Além das características dos nódulos, de acordo com a tabela Lung-Rads, a presença de enfisema pulmonar é relevante para a classificação nodular. O enfisema tem uma taxa de aparição baixa nos documentos do corpus, portanto não foi possível criar uma granularidade maior de tipos de entidade para extrair características especificas dos enfisemas descritos. Dessa forma foi criada apenas essa entidade cujo objetivo é identificar se há enfisema pulmonar descrito no documento ou não.

3.4 Anotação das entidades

O processo de anotação das entidades, que define o tipo do método como supervisionado, é um trabalho manual que idealmente deve ser feito por um ou mais especialista(s) no assunto – neste caso, médicos pneumologistas –, com conhecimento ou assessoria de um profissional da área de dados ou IA. De forma geral, esse processo vai definir a "verdade absoluta" do ponto de vista de aprendizado do modelo. Em algumas situações, essas anotações podem ser processos extremamente delicados, pois podem levar o modelo a aprender e potencializar padrões contidos no dados que sejam considerados ruins do ponto de vista da sociedade. Isso é chamado de viés (ou *bias*), e pode causar situações catastróficas dependendo do caso de uso do modelo em questão.

Para o caso deste estudo, devido ao objetivo e às limitações de esforço, tempo e custo, as anotações foram realizadas de forma manual pelo próprio autor, estando sujeito a erros considerando seu conhecimento limitado em medicina; este é um dos pontos de possíveis melhorias a ser citado no fim do trabalho.

Foram anotados e revisados, com os seis tipos de entidades mencionados anteriormente, **600 documentos** correspondente a laudos médicos de tomografia pulmonar, no decorrer de dois meses.

3.4.1 Estrutura dos dados anotados

A ferramenta spaCy espera um tipo de estrutura de dados especifico para realizar o treinamento do modelo de extração de entidades. Essa estrutura precisa conter o texto do documento em formato de *string*, e uma lista de anotações, em que cada anotação contem o tipo de entidade (*string*), e um par valores inteiros correspondentes aos índices – posição do caractere em relação ao texto – de inicio e fim da marcação, chamado de *span*.

Essa estrutura de dados, no entanto, não é amigável ao processo de marcação das entidades, pois envolveria obter manualmente as posições de inicio e fim das marcações, obrigando a fazer a contagem dos caracteres. Além disso, cada entidade marcada guarda posições fixas de inicio e fim do *span*, relativas ao texto do documento. Se o texto do documento mudar, por qualquer motivo, todas as referencias de posições marcadas a partir da posição que sofreu alteração passam a estar erradas.

Para facilitar a marcação, foi então definido uma estrutura de dados provisória e amigável ao processo, no seguinte formado: **[*tipo_da_entidade*](*span*)**.

Dado um documento como exemplo:

> *"Nódulo sólido presente no pulmão esquerdo"*

É então marcado da seguinte forma:

> "[NoduleCount](*Nódulo*) [NoduleType](*sólido*) *presente no* [NoduleLocation](*pulmão esquerdo*)"

Dessa forma, foi possível marcar as entidades de forma a não precisar se preocupar com a posição das marcações e com a corruptibilidade dos dados em caso de alguma alteração. Apesar de ser uma estrutura de dados diferente, ela contem todas as informações necessárias para seu uso no spaCy: o texto original e as várias entidades marcadas, porém com os *spans* e posições representados em outro formato.

Para contornar isso, foi escrito um programa em Python para processar os documentos marcados no formato amigável e transformá-los automaticamente no formato aceitável pelo spaCy.

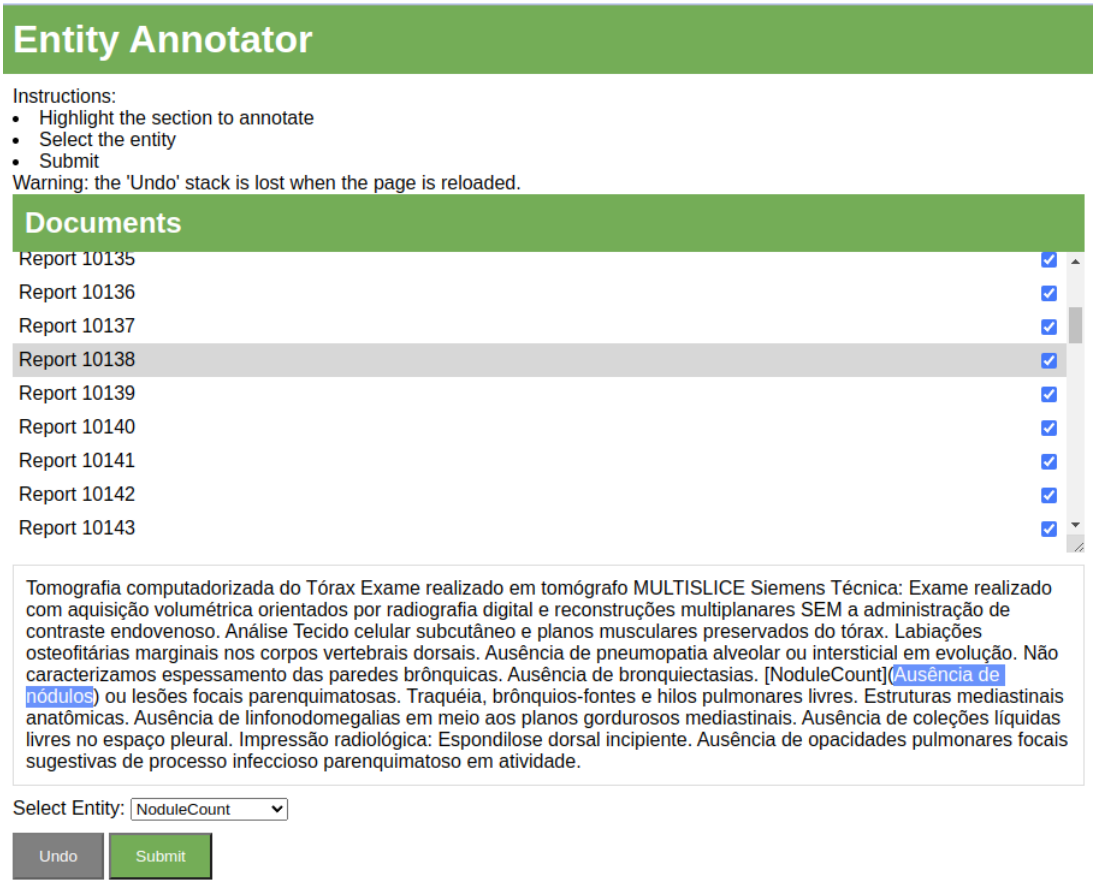
3.4.2 Aplicação auxiliar no processo de anotação

Mesmo definindo uma forma de representar as marcações amigavelmente, ao considerar centenas de documentos a serem anotados, é um processo que envolve milhares de entidades a serem marcadas e revisadas manualmente.

Para solucionar esse problema e viabilizar a marcação desses documentos em um tempo hábil, foi desenvolvido uma pequena aplicação *Web*[1] capaz de automatizar o processo de adicionar as marcações '[*tipo_da_entidade*]()' em torno dos *spans*, necessitando apenas que o usuário selecione o texto desejado, escolha a entidade e clique em um botão para marcar.

A aplicação consiste em uma tela simples listando todos os documentos carregados do servidor local. Ao escolher um dos documentos do *corpus*, seu texto aparece. Ao selecionar o texto desejado (*span*) com o mouse, o botão *Submit* fica verde e clicável, permitindo ao usuário aplicar a entidade selecionada ao texto marcado, alterando seu estado em tempo real e gravando a alteração no servidor local. Também há um seletor com as entidades disponíveis para a troca e um botão *Undo* para desfazer as ultimas alterações do documento selecionado, em caso de erro.

Figura 3.1: Exemplo da aplicação em funcionamento.



[1] https://github.com/lucasagra/TCC/tree/main/entity_annotator_app

A aplicação é simples e prática, utiliza *NodeJs* no *back-end* para servir e gravar os dados diretamente de um documento do tipo *JSON*. Na parte do *front-end* foi utilizado apenas *HTML* e *JavaScript* puro.

Dessa forma, foram marcadas mais de 4 mil entidades, conforme a tabela 3.1.

Marcações por tipo de entidade		
Tipo	**Quantidade**	**Presença**
NoduleCount	1143	1,90
NoduleLocation	1107	1,84
NoduleInfo	775	1,29
NoduleType	708	1,18
NoduleSize	450	0,75
Emphysema	96	0,16
Total	**4279**	**7.13**

Tabela 3.1: Quantidade de entidades anotadas por tipo. Presença refere-se à taxa de entidades por documento.

3.5 Processo de treinamento do modelo

Como mencionado anteriormente, o processo de treinamento envolve ensinar ao modelo pré-treinado BERTimbau como realizar a tarefa de encontrar as entidades que queremos, baseando-se nos documentos anotados conforme demonstrado acima.

O processo de treinamento de um modelo de *deep-learning* consiste em uma sequencia de etapas repetitivas, em que no caso deste estudo não é diferente:

1. O modelo recebe como entrada um lote, que é um determinado número de documentos, também chamado de *batch*;

2. O modelo tenta adivinhar quais entidades estão contidas em cada documento do lote, apenas tendo acesso aos seus textos;

3. As respostas do modelo são comparadas com as anotações das frases (*gold standard*) e é calculado um valor que determina o quão diferente são as respostas preditas das respostas anotadas, também chamado de *loss*;

4. O modelo é ajustado baseado no valor do *loss* calculado.

Todo esse processo descrito acima é denominado como um *Step*, que é repetido muitas vezes durante o treinamento e também utilizado como uma unidade de medida que referencia o quanto o modelo foi treinado. Outra unidade é a Época (*epoch*), que é nome dado quando todo o

corpus é avaliado pelo modelo uma vez. Supondo que o tamanho do lote seja de 10 documentos, e o *corpus* possua 100 documentos, uma época será concluída em 10 *steps*. Normalmente os modelos de *deep-learning* são treinados por várias épocas.

Durante o treinamento do modelo, uma das indicações de que tudo está indo bem é quando ao decorrer das épocas o valor do *loss* diminui, significando que as respostas dadas pelo modelo estão cada vez mais próximas das respostas anotadas.

Outro ponto de atenção é que durante o treinamento o modelo "vê" o mesmo documento várias vezes – dado um treinamento que possua várias épocas –, consequentemente fazendo o mesmo ajuste para um mesmo documento múltiplas vezes. Isso pode causar o que é chamado de *overfitting*: quando o modelo é super ajustado para os documentos de treinamento e consegue acertá-los, mostrando um *loss* que tende a zero, porém não consegue performar bem ao receber documentos diferentes, tornando-se inútil em seu o uso real.

Para lidar com o *overfitting* é utilizado uma pratica que envolve dividir o conjunto de dados em duas partes, uma para treinar, conforme mencionado até agora e outra parte usada exclusivamente para testar o modelo, também chamado de *dev* ou *eval*. Então, a cada *N* épocas ou *steps* o treinamento pausa e é feito uma avaliação com o conjunto de dados de teste apenas para análise, não fazendo ajuste ao modelo, em seguida o treino continua. Dessa forma é possível avaliar se o modelo realmente está aprendendo ao longo do treinamento, conseguindo extrapolar o conhecimento aprendido com o conjunto de treino para responder corretamente documentos nunca treinados.

3.5.1 Treinamento do modelo utilizando o Spacy

O spaCy fornece uma modularidade onde é possível "plugar" componentes, construindo *pipelines*. Dessa forma, o modelo NER é colocado após o *transformer* e o spaCy se encarrega de compatibilizar os dados que saem de um e entram no outro. As etapas detalhadas consistem em:

1. **Pré-processamento:** O texto de entrada é dividido em tokens (palavras, pontuações e partes de palavras) e esses tokens são convertidos em representações numéricas.

2. **Modelo Transformer:** O modelo pré-treinado baseado em *transformers* (BERT) é utilizado para gerar *embeddings* contextuais para cada token do texto de entrada. Esses *embeddings* carregam informações contextuais ricas e levam em consideração o contexto das palavras.

3. **Ajuste de dimensionalidade:** Uma camada é colocada pelo spaCy que faz um ajuste de dimensionalidade de forma que os *embeddings* do *transformer* possam ser alimentados para o próximo modelo.

4. **Modelo NER**: Diferente do que é usado comumente para tarefas de NER, em que se utilizam camadas de LSTM+CRF, o spaCy utiliza uma abordagem baseada em *Bloom embeddings* sobre CNNs (*Convolutional Neural Networks*), descritas por Lample et al. [11] como *Transition-Based Chunking Model*. O modelo recebe as representações (*embeddings*) a nível de *tokens*, em seguida as camadas convolucionais permitem agregar os vetores de representações de palavras em uma matriz contendo uma representação a nível do documento, que por sua vez é condensada para um vetor baseando-se em um mecanismo de atenção guiado por entidades, que por fim é classificado por uma etapa de multi-camadas de perceptrons.

5. **Ajuste**: O modelo é treinado usando o conjunto de dados rotulados, onde os rótulos de entidade são fornecidos para treinar o modelo a prever as entidades corretamente. Durante o treinamento, os pesos do modelo são ajustados para otimizar o desempenho na tarefa de NER. Além disso, o modelo NER envia um sinal propagado para trás (*feedback*) que também ajusta os pesos do modelo *transformer*.

Arquitetura do modelo

Para definição da arquitetura foi utilizado o sistema de reconhecimento de entidade (NER) do spaCy v3.5.2. A arquitetura do modelo consiste em um *pipeline* de dois componentes: *Transformers* (BERTimbau) + *NER*.

Configurações da arquitetura

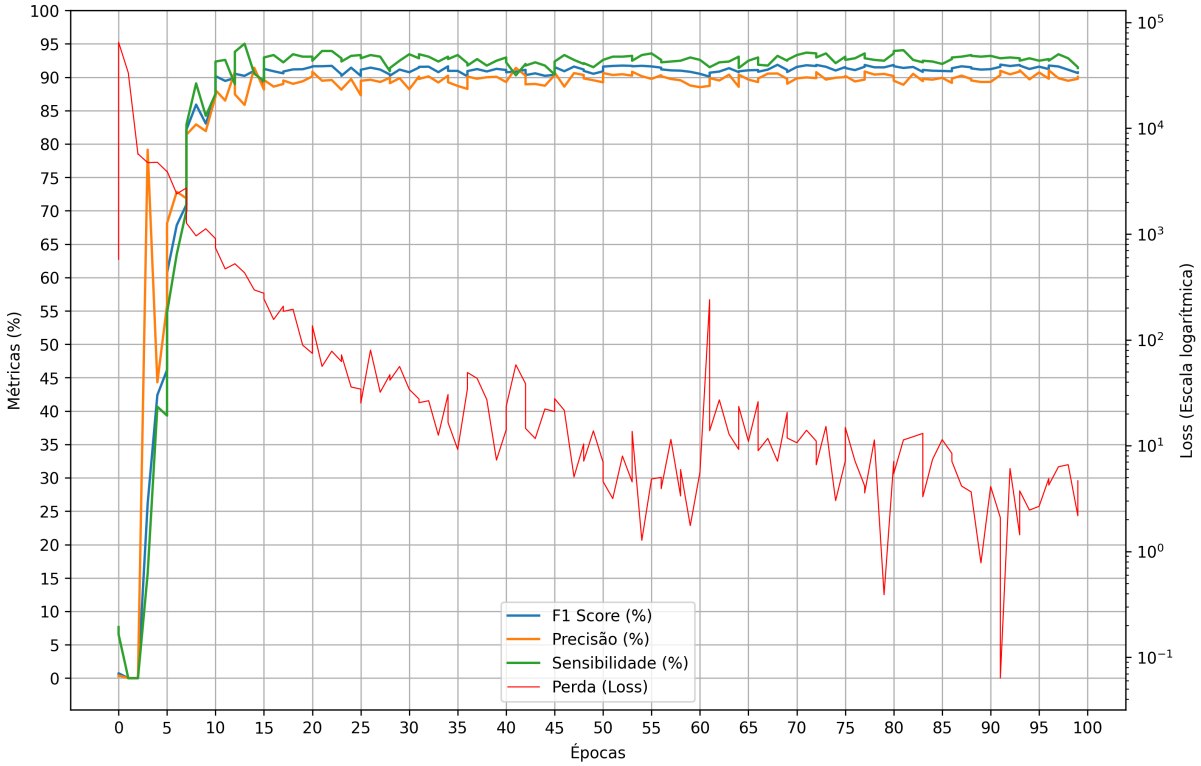
O spaCy permite centralizar toda a configuração da arquitetura em um único arquivo[2]. Dentre as principais configurações, foi utilizado lote de tamanho 128 (*batch_size*), máximo de steps igual a 20000, máximo de épocas igual a 100, frequência de avaliação a cada 20 steps, taxa de dropout de 0.1, otimizador Adam (*Adam.v1*), 250 *steps* de aquecimento e taxa de aprendizagem (*learning rate*) de 5e-5. Para o caso do modelo selecionado, o conjunto de dados foi dividido em 80% para treino (480 documentos) e 20% para teste (120 documentos).

Configurações do ambiente de treinamento

O modelo foi treinado localmente utilizando uma unidade de processamento gráfico NVIDIA RTX 3060 com 12 GB de memória dedicada, levando em torno de uma hora para completar o treinamento de 100 épocas. O procedimento[3] foi realizado em Ubuntu 20.04, utilizando Python v3.8.10, Pip v23.0.1, com o NVIDIA CUDA driver v11.2.

[2] https://github.com/lucasagra/TCC/blob/main/spacy/config.cfg
[3] https://github.com/lucasagra/TCC/tree/main/spacy

Figura 3.2: Progressão das métricas durante treinamento.



3.6 Avaliação e métricas utilizadas

A tarefa de avaliar uma predição de entidades envolve definir o que é o "correto", pois diferente de uma tarefa de classificação de intenção, por exemplo, uma tarefa NER precisa adivinhar o tipo de entidade e quais palavras do texto pertencem a ela (*span*). Para este estudo, será considerado um "acerto" as entidades que tiverem o seu tipo correto extraído e seus limites (*span*) forem iguais, formato que também é chamado de *exact match*. Ou seja, mesmo que o *span* seja parcialmente predito como a entidade correta, não será considerado um acerto, pois além do tipo, o *span* predito precisa ser igual ao *gold-standard*.

Para avaliar o modelo foi utilizada principalmente a medida F1, que considera um balanço entre as métricas de precisão e sensibilidade. A mesma foi usada como critério para escolher o ponto do treinamento em que o modelo estava em seu melhor estado, que ocorreu na época 91, conforme detalhado na tabela 4.1.

Definição das métricas

- **Precisão**: É a taxa de positivos preditos corretamente entre todos os positivos preditos.

  $Precisao = VerdadeirosPositivos/(VerdadeirosPositivos + FalsosPositivos)$

- **Sensibilidade**: É a taxa dos verdadeiros positivos preditos entre o que realmente foi marcado.

$$Sensibilidade = VerdadeirosPositivos/(VerdadeirosPositivos + FalsosNegativos)$$

- **Medida F1**: Função de precisão e sensibilidade que permite fazer um balanço entre as duas métricas.

  $$MedidaF1 = 2 * Precisao * Sensibilidade/(Precisao + Sensibilidade)$$

No contexto de NER, verdadeiros positivos são entidades que foram preditas corretamente (*exact match*), falsos positivos são entidades que foram preditas erroneamente, e falsos negativos são entidades que deveriam ser preditas mas não foram. É valido ressaltar que o falso positivo faz referência a entidade predita e o falso negativo faz referência a entidade anotada, portanto, ao acontecer uma predição errada, ambos os casos podem acontecer simultaneamente. Por exemplo, supondo um outro sistemas de entidades em que:

Anotado:

"O [médico] atendeu o [paciente] com [dor de cabeça]."

Predito:

"O [médico] [atendeu] o paciente com [dor] de cabeça."

Tem-se:

- Verdadeiro Positivo (1): [*médico*] (supondo que o tipo foi predito corretamente)
- Falso Positivo (2): [*atendeu*, *dor*]
- Falso Negativo (2): [*paciente*, *dor de cabeça*]

4

Resultados

Com a metodologia descrita acima, foi possível obter um modelo com 91,89% de medida F1, detalhado na tabela 4.1.

Tabela 4.1: Métricas de precisão, sensibilidade e medida F1, por entidade, do modelo selecionado (maior medida F1). Suporte representa a quantidade de anotações; Variação aponta a quantidade de diferentes *tokens* anotados em cada tipo.

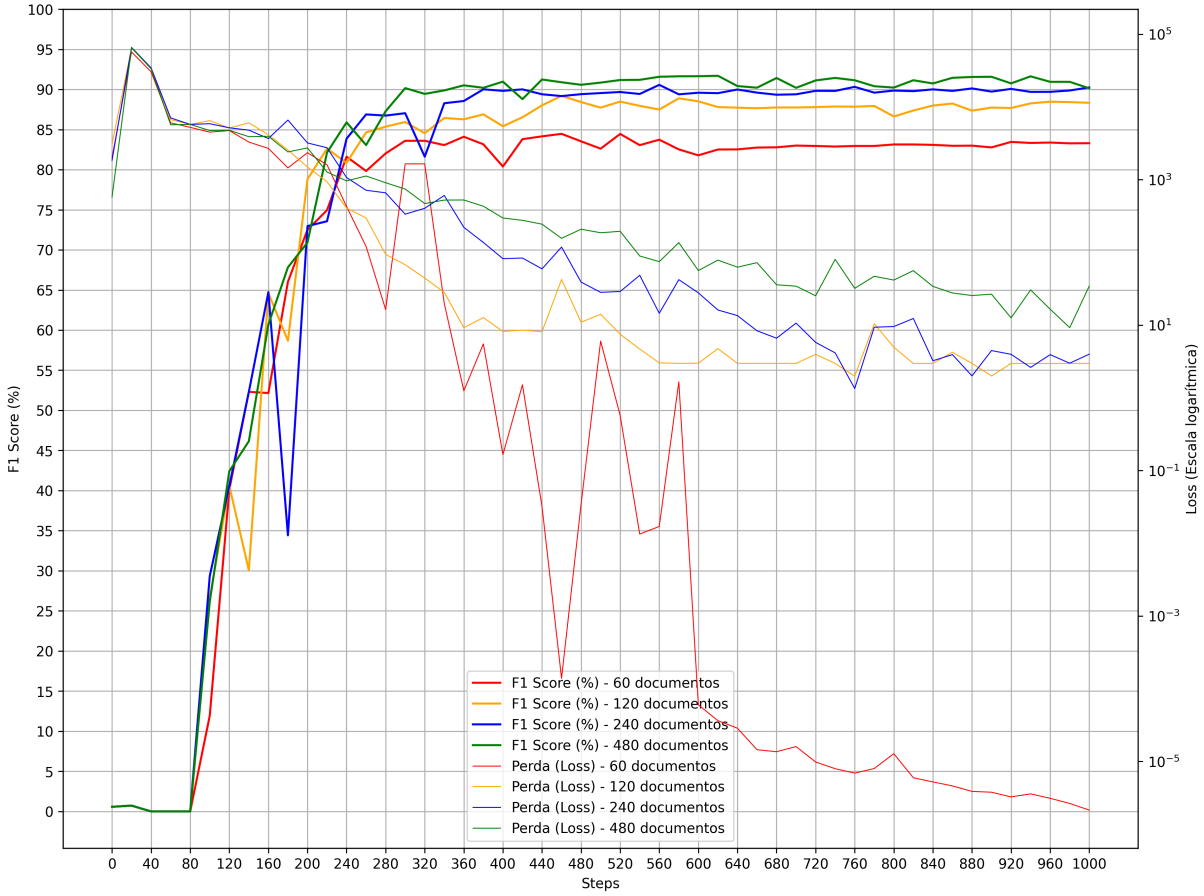
Métricas por tipo de entidade					
Tipo	Precisão	Sensibilidade	Medida F1	Suporte	Variação
NoduleCount	0.9476	0.9518	0.9497	1143	110
NoduleType	0.9848	0.9630	0.9738	708	44
NoduleLocation	0.8506	0.9071	0.8779	1107	147
NoduleSize	0.9556	0.9663	0.9609	450	9[1]
NoduleInfo	0.8519	0.8712	0.8614	775	167
Emphysema	0.8571	0.8571	0.8571	96	48
Total	**0.9096**	**0.9284**	**0.9189**	**4279**	*

Analisando por tipo de entidade, é notável que o tipo *Emphysema* tem um deficit de exemplos de suporte, o que explica sua baixa performance; mas também é possível relacionar que os tipos *NoduleInfo* e *NoduleLocation*, possuindo variações maiores tendem a ter uma pontuação menor, necessitando de uma maior taxa de suporte para se saírem bem.

Foi feita também uma análise da importância da quantidade de documentos a serem usados no conjunto de treino, executando o processo de treinamento por 1000 *steps* com quantidades variadas de documentos (60, 120, 240, 480), avaliando-os com o mesmo conjunto de dados (120 documentos), o que resultou no gráfico da figura 4.1.

[1] A quantidade de diferentes *tokens* de *NoduleSize* real é 136, porém em sua grande maioria são valores numéricos, que para o conceito de "Variação" fazer sentido, foram agregados na tabela como um único valor.

Figura 4.1: Comportamento das métricas em relação ao número de documentos utilizados no treinamento.



Observa-se dois pontos interessantes: o primeiro é que mesmo com uma quantidade inicial pequena de documentos (60) já é possível obter um modelo consideravelmente bom com quase 85% de medida F1, graças aos *embeddings* pre-treinados do BERT; o segundo ponto é que a quantidade de documentos permanece como um fator de importancia, sendo possível aproximar-se dos 95% ao extrapolar os resultados do gráfico para 1000 ou 2000 documentos.

Sugimoto et al. [20] realizou um estudo semelhante onde foi obtido uma medida F1 de 95.36%, o que é condizente com o resultado obtido neste estudo considerando as diferenças existentes entre ambos:

1. Foi utilizado um modelo pré-treinado BERT(Wiki) em japonês com dados da Wikipedia;

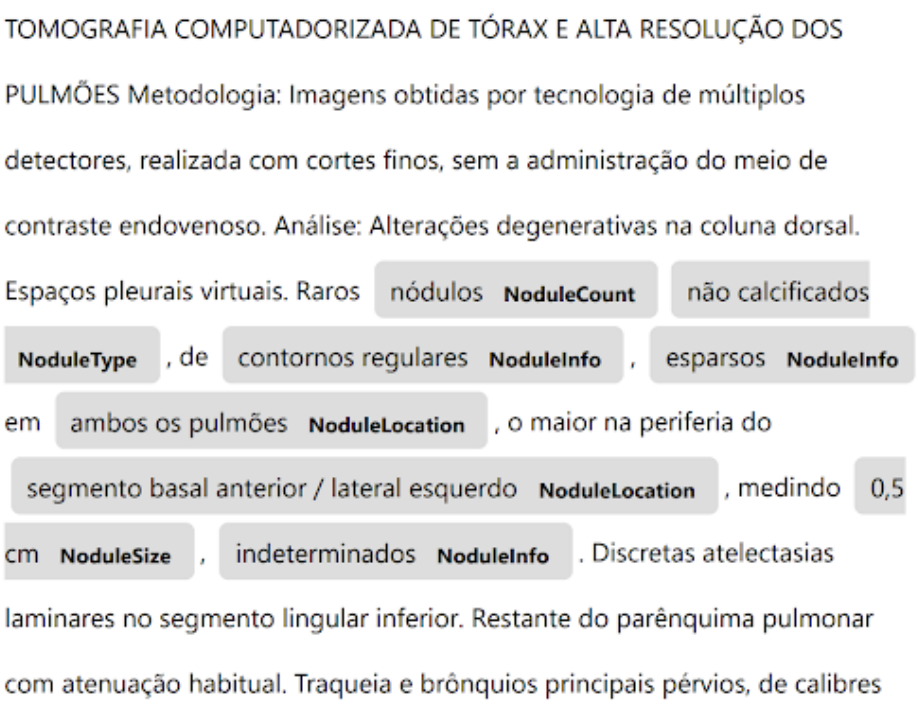
2. No estudo citado, o modelo BERT(Wiki) passou por uma fase intermediária adicional de pré-treinamento, sendo submetido a um corpus cedido por um hospital contendo 952 mil documentos clínicos, melhorando sua contextualização nesse escopo. Essa etapa extra modificou o resultado final de 94.32% para 95.36%;

3. O corpus de treinamento NER, contendo as entidades, foi cerca de 3 vezes maior que o

deste estudo, contendo 13699 entidades anotadas. Para garantir o *gold-standard*, o corpus foi anotado separadamente por três especialistas e suas anotações comparadas entre si.

4. Para a etapa NER foi utilizado a arquitetura BiLSTM-CRF.

Figura 4.2: Exemplos de entidades sendo anotadas pelo modelo treinado em um documento do conjunto de teste.



4.1 Discussão dos desafios enfrentados e possíveis melhorias

A definição e anotação das entidades é uma tarefa particularmente desafiadora pois exige uma sincronia e padronização de como o *gold-standard* deve ser definido e cada tipo de entidade deve ser anotada. Esses requisitos ficam ainda mais difíceis quando se tem mais de uma pessoa compartilhando essa tarefa, o que é necessário dado que uma grande quantidade de dados é geralmente desejável e há um limite de quanto uma pessoa consegue fazer em um determinado tempo. Levando em consideração a avaliação em formato de *exact match*, por exemplo, é comum haver discordância entre os anotadores do *span* anotado, e isso afetaria a avaliação do modelo. Portanto é comum os anotadores revisarem as anotações dos pares e até trazerem uma pessoa de fora para lidar com os casos divergentes de forma imparcial.

Nesse âmbito há melhorias claras a serem sugeridas para este trabalho como a utilização de um conjunto de dados maior e dados que sejam anotados ou revisados por um ou mais profissionais da área.

Uma outra dificuldade é a pequena quantidade de publicações na área quando se trata especificamente do idioma português brasileiro. Névéol et al. [13] aponta que dentro do universo das publicações de estudos relacionados aos termos “PubMed” e “PLN”, desconsiderando o inglês, apenas 3% (14 de 435) dessas são referentes à língua portuguesa, ficando atrás de idiomas como o sueco, holandês e japonês.

Seguindo o resultado obtido por Sugimoto et al. [20], uma outra possível melhoria seria utilizar um modelo BERT pré-treinado em contexto médico português, dando mais ênfase aos termos clínicos presentes nos documentos em formato de laudo médico.

5

Conclusão

No estudo foi demonstrado um método para treinar um modelo de informação especializado em identificar entidades personalizadas, definidas in-loco para extrair as principais características de nódulos pulmonares de laudos médicos. Foi obtido um modelo com 91,89% de medida F1 considerando um investimento de recursos e tempo relativamente baixo. Isso foi possível devido à utilização de uma arquitetura de modelo baseado em modelos de linguagens pré-treinados, o que reduz a quantidade de dados próprios necessários para se ter um bom resultado ao reaproveitar modelos já treinados pela comunidade acadêmica; e ao fato de utilizar uma ferramenta que abstrai grande parte da complexidade estrutural do *deep-learning*, sendo possível direcionar a maior parte do esforço investido no tratamento dos dados de treinamento.

Os resultados obtidos mostram que o método é reproduzível para diferentes casos de uso, sendo possível bons resultados utilizando até menos dados anotados do que o demonstrado aqui, desde que os mesmos sejam auditados por um especialista na área.

5.1 Possíveis direções futuras de pesquisa

O principal objetivo do estudo foi extrair valor dos dados não estruturados no âmbito de sistemas digitais, definindo e identificando as entidades que seriam uteis para a classificação nodular de acordo com a tabela Lung-Rads. Dessa forma, uma sugestão de futura pesquisa é utilizar as entidades nodulares extraídas para treinar modelos de classificação que busquem classificar o grau de malignidade do nódulo descrito, elevando ainda mais a utilidade prática do modelo aqui definido.

Outras possíveis direções de estudos futuros seriam treinar um modelo semelhante utilizando a arquitetura BiLSTM-CRF, substituindo a arquitetura fornecida pelo spaCy, e comparar os resultados obtidos, visto que a literatura disponível de modelos em português no mesmo contexto deixa a desejar.

Dado que o modelo deste estudo foi treinado para reconhecer características nodulares, há uma boa probabilidade que ele também funcione bem considerando nódulos em outros contextos clínicos, fora da área pneumológica. Avaliar essa questão possibilitaria ter uma melhor estimativa de até que ponto o modelo é capaz de extrapolar seu aprendizado para outros contextos, talvez até eliminando a necessidade de construir conjuntos de dados específicos e retreinar outros modelos para tarefas semelhantes.

Referências bibliográficas

[1] Abhimanyu S Ahuja. The impact of artificial intelligence in medicine on the future role of the physician. *PeerJ*, 7:e7702, October 2019.

[2] Zaid Alyafeai, Maged Saeed AlShaibani, and Irfan Ahmad. A survey on transfer learning in natural language processing, 2020.

[3] A R Aronson. Effective mapping of biomedical text to the UMLS metathesaurus: the MetaMap program. *Proc. AMIA Symp.*, pages 17–21, 2001.

[4] Sean Blandin Knight, Phil A. Crosbie, Haval Balata, Jakub Chudziak, Tracy Hussell, and Caroline Dive. Progress and prospects of early detection in lung cancer. *Open Biology*, 7 (9):170070, 2017. **DOI** 10.1098/rsob.170070. URL https://royalsocietypublishing.org/doi/abs/10.1098/rsob.170070.

[5] Olivier Bodenreider. The unified medical language system (UMLS): integrating biomedical terminology. *Nucleic Acids Res.*, 32(Database issue):D267–70, January 2004.

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

[7] Andre Esteva, Alexandre Robicquet, Bharath Ramsundar, Volodymyr Kuleshov, Mark DePristo, Katherine Chou, Claire Cui, Greg Corrado, Sebastian Thrun, and Jeff Dean. A guide to deep learning in healthcare. *Nature Medicine*, 25(1):24–29, January 2019.

[8] Melissa Rudy Fox News. First ai-generated drug enters human clinical trials, targeting chronic lung disease patients, 2023. URL https://nypost.com/2023/06/29/first-ai-generated-drug-enters-human-clinical-trials-targeting-chronic-lung-dis

[9] Matthew Honnibal, Ines Montani, Sofie Van Landeghem, and Adriane Boyd. spacy: Industrial-strength natural language processing in python. 2020. **DOI** 10.5281/zenodo.1212303. URL https://spacy.io/.

[10] Srinidhi Karthikeyan, Alba G. Seco de Herrera, Faiyaz Doctor, and Asim Mirza. An ocr post-correction approach using deep learning for processing medical reports. *IEEE Transactions on Circuits and Systems for Video Technology*, 32(5):2574–2581, 2022. **DOI** 10.1109/TCSVT.2021.3087641.

[11] Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. Neural architectures for named entity recognition, 2016.

[12] Amil Merchant, Elahe Rahimtoroghi, Ellie Pavlick, and Ian Tenney. What happens to BERT embeddings during fine-tuning? In *Proceedings of the Third BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*, pages 33–44, Online, November 2020. Association for Computational Linguistics. **DOI** 10.18653/v1/2020.blackboxnlp-1.4. URL https://aclanthology.org/2020.blackboxnlp-1.4.

[13] Aurélie Névéol, Hercules Dalianis, Sumithra Velupillai, Guergana Savova, and Pierre Zweigenbaum. Clinical natural language processing in languages other than english: opportunities and challenges. *J. Biomed. Semantics*, 9(1):12, March 2018.

[14] American College of Radiology Committee on Lung-RADS®. Lung-rads assessment categories 2022, 2022. URL https://www.acr.org/-/media/ACR/Files/RADS/Lung-RADS/Lung-RADS-2022.pdf.

[15] Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations, 2018.

[16] Yuqi Si, Jingqi Wang, Hua Xu, and Kirk Roberts. Enhancing clinical concept extraction with contextual embeddings. *Journal of the American Medical Informatics Association*, 26(11):1297–1304, jul 2019. **DOI** 10.1093/jamia/ocz096. URL https://doi.org/10.1093%2Fjamia%2Focz096.

[17] Jenni A. M. Sidey-Gibbons and Chris J. Sidey-Gibbons. Machine learning in medicine: a practical introduction. *BMC Medical Research Methodology*, 19(1):64, Mar 2019. ISSN 1471-2288. **DOI** 10.1186/s12874-019-0681-4. URL https://doi.org/10.1186/s12874-019-0681-4.

[18] C. Isabela S. Silva, Edson Marchiori, Arthur Soares Souza Júnior, and Nestor L. Müller. Consenso brasileiro ilustrado sobre a terminologia dos descritores e padrões fundamentais da tc de tórax. *Jornal Brasileiro de Pneumologia*, 36(1):99–123, Jan 2010. ISSN 1806-3713. **DOI** 10.1590/S1806-37132010000100016. URL https://doi.org/10.1590/S1806-37132010000100016.

[19] Fábio Souza, Rodrigo Nogueira, and Roberto Lotufo. BERTimbau: pretrained BERT models for Brazilian Portuguese. In *9th Brazilian Conference on Intelligent Systems, BRACIS, Rio Grande do Sul, Brazil, October 20-23 (to appear)*, 2020.

[20] Kento Sugimoto, Toshihiro Takeda, Jong-Hoon Oh, Shoya Wada, Shozo Konishi, Asuka Yamahata, Shiro Manabe, Noriyuki Tomiyama, Takashi Matsunaga, Katsuyuki Nakanishi, and Yasushi Matsumura. Extracting clinical terms from radiology reports with deep learning. *Journal of Biomedical Informatics*, 116:103729, 2021. ISSN 1532-0464. **DOI** https://doi.org/10.1016/j.jbi.2021.103729. URL https://www.sciencedirect.com/science/article/pii/S1532046421000587.

[21] Hyuna Sung, Jacques Ferlay, Rebecca L. Siegel, Mathieu Laversanne, Isabelle Soerjomataram, Ahmedin Jemal, and Freddie Bray. Global cancer statistics 2020: Globocan estimates of incidence and mortality worldwide for 36 cancers in 185 countries. *CA: A Cancer Journal for Clinicians*, 71(3):209–249, 2021. **DOI** https://doi.org/10.3322/caac.21660. URL https://acsjournals.onlinelibrary.wiley.com/doi/abs/10.3322/caac.21660.

[22] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need, 2017.

[23] Jody Vykoukal, Johannes F Fahrmann, Nikul Patel, Masayoshi Shimizu, Edwin J Ostrin, Jennifer B Dennison, Cristina Ivan, Gary E Goodman, Mark D Thornquist, Matt J Barnett, Ziding Feng, George A Calin, and Samir M Hanash. Contributions of circulating microRNAs for early detection of lung cancer. *Cancers (Basel)*, 14(17), August 2022.

[24] Jorge Wagner, Rodrigo Wilkens, Marco Idiart, and Aline Villavicencio. The brwac corpus: A new open resource for brazilian portuguese. 05 2018.

[25] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. Huggingface's transformers: State-of-the-art natural language processing, 2020.

[26] Cecilia Zappa and Shaker A. Mousa. Non-small cell lung cancer: current treatment and future advances. *Translational Lung Cancer Research*, 5(3), 2016. ISSN 2226-4477. URL https://tlcr.amegroups.org/article/view/8139.

[27] Qing T. Zeng, Sergey Goryachev, Scott Weiss, Margarita Sordo, Shawn N. Murphy, and Ross Lazarus. Extracting principal diagnosis, co-morbidity and smoking status for asthma research: evaluation of a natural language processing system. *BMC Medical*

Informatics and Decision Making, 6(1):30, Jul 2006. ISSN 1472-6947.
DOI 10.1186/1472-6947-6-30. URL https://doi.org/10.1186/1472-6947-6-30.

[28] Fuzhen Zhuang, Zhiyuan Qi, Keyu Duan, Dongbo Xi, Yongchun Zhu, Hengshu Zhu, Hui Xiong, and Qing He. A comprehensive survey on transfer learning, 2020.