UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE INFORMÁTICA
CURSO DE ENGENHARIA DE COMPUTAÇÃO

JONATAS TSCHÁ SANTORO

Uma ferramenta de processamento de linguagem natural para extração de dados em prescrições médicas eletrônicas

Monografia apresentada como requisito parcial para a obtenção do grau de Bacharel em Engenharia da Computação

Orientador: Prof. Dr. Anderson Tavares

Porto Alegre
2023

“Success is the sum of small efforts,
repeated day-in and day-out.”
— ROBERT COLLIER

AGRADECIMENTOS

Gostaria de agradecer a minha família e amigos, que me apoiaram incondicionalmente ao longo de toda a minha trajetória acadêmica. Sem o amor, a paciência e a compreensão de vocês, este trajeto seria mais difícil.

Agradeço em especial aos meus pais, que sempre estiveram ao meu lado, me encorajando a perseguir os meus sonhos e me dando todo o suporte necessário para alcançá-los. Agradeço também a minha irmã, que me inspira com o seu exemplo de dedicação e determinação.

Aos meus amigos, agradeço por me acompanharem em momentos de alegria e de dificuldade. Sou muito grato por ter vocês em minha vida.

Ao meu orientador, agradeço pela dedicação e orientação durante todo o processo de elaboração deste trabalho.

Por fim, gostaria de agradecer aos professores e demais profissionais da Universidade Federal do Rio Grande do Sul que contribuíram para a minha formação, compartilhando conhecimento e estimulando meu desenvolvimento pessoal e profissional.

RESUMO

Este trabalho apresenta uma ferramenta de processamento de linguagem natural, destinada à extração de dados em prescrições médicas eletrônicas. O principal objetivo é proporcionar eficiência e praticidade, permitindo a extração automática de informações relevantes contidas nas prescrições, como medicamento, concentração, dosagem, frequência, prazo e observações. Para atingir este objetivo, amostras de prescrições médicas eletrônicas no formato PDF foram coletadas, pré-processadas e rotuladas com as entidades requeridas, permitindo a geração de novas amostras. Posteriormente, procedeu-se ao ajuste fino de um modelo de reconhecimento de entidades nomeadas (NER), utilizando a biblioteca de processamento de linguagem natural SpaCy. Com o auxílio do modelo treinado, desenvolveu-se uma interface de programação de aplicação (API) que emprega o modelo para identificar as entidades nas prescrições médicas e conduz um pós-processamento mediante regras lógicas e expressões regulares, visando a validação das entidades detectadas e a padronização dos resultados. O resultado final consiste nas entidades extraídas e suas informações correspondentes, com a possibilidade de ser prontamente aproveitado em aplicativos de saúde através do uso da API.

Palavras-chave: Aprendizado de máquina. Processamento de linguagem natural. Prescrições médicas eletrônicas. Reconhecimento de entidades nomeadas.

A natural language processing tool for data extraction from electronic medical prescriptions

ABSTRACT

This paper presents a natural language processing tool designed for data extraction from electronic medical prescriptions. The main goal is to provide efficiency and convenience, allowing for the automatic extraction of relevant information contained in prescriptions, such as medication, concentration, dosage, frequency, duration, and observations. To achieve this goal, electronic medical prescription samples in PDF format were collected, pre-processed, and labeled with the required entities, allowing for the generation of new samples. Subsequently, fine-tuning of a named entity recognition (NER) model was performed using the SpaCy natural language processing library. With the assistance of the trained model, an application programming interface (API) was developed that employs the model to identify entities in medical prescriptions and conducts post-processing through logical rules and regular expressions, aiming to validate the detected entities and standardize the results. The final outcome consists of the extracted entities and their corresponding information, with the possibility of being readily utilized in health applications through the use of the API.

Keywords: Machine learning. Natural language processing. Electronic medical prescriptions. Named entity recognition.

LISTA DE ABREVIATURAS E SIGLAS

NER	Named Entity Recognition
PLN	Processamento de Linguagem Natural
API	Application Programming Interface
RNN	Rede Neural Recorrente
LSTM	Rede de Memória de Longo Prazo

LISTA DE FIGURAS

Figura 2.1 Visão geral das categorias de aprendizado de máquina 13
Figura 2.2 Exemplo de matriz de confusão multiclasse, com frequência normalizada .. 17
Figura 2.3 Pipelines de processamento de linguagem do SpaCy 18

Figura 4.1 Fluxograma da metodologia aplicada 25
Figura 4.2 Texto extraído do HTML representado na Figura A.2 26
Figura 4.3 Exemplo de rotulagem das amostras com a ferramenta UBIAI 27
Figura 4.4 Exemplo de arquivo JSON exportado pela ferramenta UBIAI após rotulagem das amostras 27
Figura 4.5 Amostra gerada a partir da amostra representada na Figura 4.2 28
Figura 4.6 Diagrama de sequência da API desenvolvida 30

Figura 6.1 Métricas dos Modelos 1, 2 e 3 36
Figura 6.2 F1-Score das entidades dos modelos 1, 2 e 3 37
Figura 6.3 Métricas dos modelos 1, 10 e 19. A escala do eixo *Score* se inicia em 0.6... 38
Figura 6.4 F1-Score das entidades dos modelos 1, 10 e 19 39
Figura 6.5 Métricas dos modelos 2, 11 e 20. A escala do eixo *Score* se inicia em 0.6... 39
Figura 6.6 F1-Score das entidades dos modelos 2, 11 e 20 40
Figura 6.7 Métricas dos modelos 3, 12 e 21 40
Figura 6.8 Métricas dos modelos 1, 4 e 7 42
Figura 6.9 Métricas dos modelos 2, 5 e 8 42
Figura 6.10 Métricas dos modelos 3, 6 e 9 43
Figura 6.11 F1-Score dos modelos selecionados 44
Figura 6.12 F1-Score das entidade dos modelos selecionados 45
Figura 6.13 Matriz de confusão do modelo 3 46
Figura 6.14 Matriz de confusão do modelo 18 47
Figura 6.15 Exemplo 1 - Texto extraído 47
Figura 6.16 Exemplo 1 - Informações retornadas pela API 48
Figura 6.17 Exemplo 2 - Texto extraído 48
Figura 6.18 Exemplo 2 - Informações retornadas pela API 48

Figura A.1 Prescrição médica eletrônica emitida em PDF 54
Figura A.2 HTML convertido da Prescrição médica eletrônica emitida em PDF 55

Figura A.1 Exemplo 1 - Prescrição médica emitida pelo Unimed 57
Figura A.2 Exemplo 2 - Prescrição médica emitida pela plataforma MEMED 58

LISTA DE TABELAS

Tabela 6.1 Modelos selecionados para a avaliação do desempenho em relação ao grupo de amostras....36
Tabela 6.2 Modelos selecionados para a avaliação do desempenho em relação ao *dropout*....38
Tabela 6.3 Modelos selecionados para a avaliação do desempenho em relação ao *batch size*....41
Tabela 6.4 Modelos selecionados para treinamento final....43

Tabela B.1 Configurações dos modelos - os grupos se referem ao grupos de amostras, conforme descrito na subseção 5.3....56

SUMÁRIO

1 INTRODUÇÃO **11**
2 FUNDAMENTAÇÃO TEÓRICA **13**
2.1 Aprendizado de Máquina **13**
2.1.1 Avaliação de Modelos 14
2.1.2 Métricas de Avaliação e Matriz de Confusão 15
2.2 Processamento de Linguagem Natural **17**
2.2.1 Técnicas de Extração de Informação 17
2.2.2 SpaCy 18
2.2.3 *Fine-Tuning* 19
2.3 Privacidade e ética em dados médicos **20**
3 TRABALHOS RELACIONADOS E ANÁLISE DE MERCADO **21**
3.1 Trabalhos Relacionados **21**
3.1.1 *Neural Architectures for Named Entity Recognition* 21
3.1.2 *CADEC: A Corpus of Adverse Drug Event Annotations* 22
3.1.3 *Introduction to the CoNLL-2003 Shared Task* 23
3.2 Análise de Mercado **24**
4 METODOLOGIA **25**
4.1 Coleta e pré-processamento das amostras **25**
4.1.1 Coleta das amostras 25
4.1.2 Pré-processamento das amostras 26
4.2 Rotulagem das amostras **27**
4.3 Geração de amostras **27**
4.3.1 Grupo de amostras gerado a partir de dados originais 28
4.3.2 Grupo de amostras gerado a partir da variação de dados originais 28
4.4 Desenvolvimento da API **29**
5 EXECUÇÃO DO *PIPELINE* **31**
5.1 Configuração do modelo **31**
5.2 Preparação dos dados **31**
5.3 Validação cruzada **32**
5.4 Avaliação dos modelos **33**
5.5 Treinamento final dos modelos selecionados **34**
6 RESULTADOS **35**
6.1 Avaliação dos modelos de validação cruzada **35**
6.1.1 Desempenho em relação ao grupo de amostras 36
6.1.2 Desempenho em relação ao *dropout* 37
6.1.3 Desempenho em relação ao *batch size* 41
6.1.4 Seleção dos modelos para treinamento final 43
6.2 Avaliação dos modelos selecionados em conjunto de teste **44**
6.3 Testes da API **47**
7 DISCUSSÃO **49**
7.1 Análise crítica dos resultados **49**
7.2 Limitações e possibilidades de melhoria **49**
8 CONCLUSÃO **51**
REFERÊNCIAS **52**
APÊNDICE A — PRÉ-PROCESSAMENTO **54**
APÊNDICE B — CONFIGURAÇÕES DOS MODELOS **56**
ANEXO A — EXEMPLOS DE AMOSTRAS COLETADAS **57**

1 INTRODUÇÃO

A utilização de tecnologias avançadas de inteligência artificial tem se tornado cada vez mais comum em diversos setores, oferecendo soluções mais eficientes e práticas para as mais diversas necessidades. A implementação dessas tecnologias tem se mostrado um fator essencial para o sucesso de muitas empresas, que buscam se destacar em um mercado cada vez mais competitivo.

O uso da tecnologia tem se expandido significativamente desde 2017, mas com um crescimento mais sutil nos últimos anos. Embora a adoção da inteligência artificial tenha mais do que dobrado desde 2017, a proporção de organizações que utilizam a tecnologia estabilizou-se entre 50 e 60 por cento. No entanto, um grupo de empresas líderes que obtêm os maiores retornos financeiros com a inteligência artificial continuam a se destacar dos seus concorrentes. Em suma, a inteligência artificial tem enorme importância para as empresas que desejam se destacar no mercado (CHUI et al., 2022).

As empresas que afirmam terem implementado ativamente o uso de inteligência artificial em operações comerciais representam 42%, e 34% afirmam estarem explorando a tecnologia. O principal foco está na automação de processos de TI (33%), segurança e detecção de ameaças (29%), automação de processos de negócios (28%) e inteligência de mercado (26%) (IBM Corporation, 2022).

Nesse contexto, a Zelle, empresa incubada no Centro de Empreendimentos em Informática (CEI) da Universidade Federal do Rio Grande do Sul (UFRGS), encontrou a necessidade de incluir em sua aplicação, que tem como objetivo ajudar pais e mães a manterem uma troca de informações de forma equilibrada e eficiente sobre os filhos (ZELLE, 2023), a extração de dados de prescrições médicas eletrônicas, com o objetivo de melhorar a experiência dos usuários.

A prescrição médica eletrônica é uma tendência crescente na área da saúde, oferecendo diversas vantagens em relação às prescrições em papel, como maior segurança, melhor controle de estoque e menor chance de erros de dosagem e administração. No entanto, a extração de dados dessas prescrições pode ser um processo manual e demorado.

Ao perceber a necessidade da Zelle e a crescente utilização da inteligência artificial em todo o mundo, identificou-se uma oportunidade para criar uma ferramenta que atendesse às demandas da Zelle. Por meio da aplicação de técnicas de aprendizado de máquina e processamento de linguagem natural, a ferramenta desenvolvida é capaz de identificar informações relevantes presentes em prescrições médicas, como medicamento,

concentração, dosagem, frequência, prazo e observação. Isso oferece uma solução mais prática e eficiente para a importação dessas informações para o aplicativo da Zelle, o que pode melhorar a experiência do usuário.

Além de atender às necessidades específicas da empresa, a ferramenta desenvolvida tem potencial para se tornar uma aplicação independente, oferecendo uma solução útil para uma variedade de empresas que precisam lidar com informações de prescrições médicas eletrônicas. Desse modo, a ferramenta desenvolvida é capaz de oferecer mais praticidade e agilidade aos usuários.

2 FUNDAMENTAÇÃO TEÓRICA

2.1 Aprendizado de Máquina

O aprendizado de máquina é um subcampo da inteligência artificial que busca desenvolver algoritmos e modelos capazes de aprender a partir de dados e melhorar seu desempenho ao longo do tempo (MITCHELL, 1997). Esses algoritmos podem ser divididos em categorias, como aprendizado supervisionado, não supervisionado e por reforço, dependendo da natureza dos dados de entrada e do objetivo do aprendizado (BISHOP, 2006). A Figura 2.1 apresenta uma visão geral destas categorias:

Figura 2.1 – Visão geral das categorias de aprendizado de máquina

Fonte: (AQUARELA, 2022)

Neste trabalho, utilizou-se o aprendizado supervisionado para treinar um modelo de reconhecimento de entidades nomeadas (NER, do inglês *Named Entity Recognition*).

O aprendizado supervisionado é um tipo de aprendizado de máquina em que um modelo é treinado com base em um conjunto de dados rotulados, ou seja, dados que contêm tanto os exemplos de entrada quanto os resultados desejados (GOODFELLOW; BENGIO; COURVILLE, 2016).

Para o aprendizado supervisionado, utilizou-se redes neurais. O SpaCy, biblioteca utilizada para o desenvolvimento deste trabalho, integra o uso de redes neurais em várias

tarefas de NLP, incluindo reconhecimento de entidades nomeadas. O treinamento de uma rede neural envolve o ajuste dos parâmetros (pesos e viés) da rede para minimizar o erro de predição nos dados de treinamento.

2.1.1 Avaliação de Modelos

Overfitting é um problema comum em aprendizado de máquina, especialmente em modelos complexos como redes neurais. Ocorre quando um modelo se ajusta muito bem aos dados de treinamento, mas não generaliza bem para dados não vistos. Em outras palavras, o modelo aprende padrões específicos dos dados de treinamento que não são representativos do processo gerador subjacente (BISHOP, 2006). Isso geralmente ocorre quando o modelo é muito complexo ou tem muitos parâmetros em relação ao tamanho do conjunto de dados disponível.

Para prevenir o *overfitting*, são utilizados parâmetros que não são aprendidos durante o treinamento do modelo, mas sim definidos previamente. Esses parâmetros recebem o nome de hiperparâmetros. Eles têm um impacto significativo na performance dos algoritmos de aprendizado de máquina e podem influenciar fatores como a velocidade de convergência, a capacidade de generalização e a complexidade do modelo (HUTTER; KOTTHOFF; VANSCHOREN, 2019). Ajustar os hiperparâmetros corretamente é importante para evitar *overfitting* e melhorar a capacidade de generalização do modelo.

No contexto deste trabalho, é importante explorar e otimizar os hiperparâmetros do modelo NER para garantir que o modelo treinado seja eficiente e eficaz na extração de informações relevantes das prescrições médicas. Abaixo, seguem os principais hiperparâmetros utilizados neste trabalho:

- *Dropout*: *Dropout* é uma técnica de regularização usada para evitar o *overfitting* em redes neurais. Consiste em desligar aleatoriamente algumas unidades (neurônios) durante o treinamento, o que reduz a dependência entre os neurônios e ajuda a evitar a coadaptação excessiva ao conjunto de treinamento (SRIVASTAVA et al., 2014). Desta forma, a taxa de *dropout* é um hiperparâmetro que determina a proporção de neurônios que serão desativados durante o treinamento. Uma taxa de *dropout* bem ajustada pode melhorar a capacidade de generalização do modelo NER.
- *Batch size*: O tamanho do *batch* (ou lote) é um hiperparâmetro que determina o número de exemplos de treinamento processados de uma só vez durante uma única

época do algoritmo de otimização. O uso de *mini-batches* pode acelerar o processo de treinamento e torná-lo mais eficiente em termos de memória em comparação com o processamento de todos os exemplos de uma vez (GOODFELLOW; BENGIO; COURVILLE, 2016). No entanto, o tamanho do *batch* pode influenciar a qualidade do aprendizado, e é importante encontrar um equilíbrio entre a eficiência computacional e a capacidade de generalização do modelo.

- Épocas: As épocas se referem ao número de vezes que o algoritmo de otimização percorre todo o conjunto de treinamento. Cada época é composta por um número de passos (ou atualizações de peso) igual ao tamanho do conjunto de treinamento dividido pelo tamanho do *batch*. Um número maior de época pode levar a um melhor desempenho do modelo, mas também pode aumentar o risco de *overfitting* e o tempo de treinamento (GOODFELLOW; BENGIO; COURVILLE, 2016).
- Taxa de aprendizado: A taxa de aprendizado é um hiperparâmetro que controla a magnitude das atualizações dos pesos durante o treinamento. Uma taxa de aprendizado alta pode levar a uma convergência mais rápida, mas também pode causar oscilações e instabilidade nos pesos. Por outro lado, uma taxa de aprendizado muito baixa pode resultar em convergência lenta e um modelo subótimo (BOTTOU, 2010). A escolha adequada da taxa de aprendizado é crucial para o sucesso do treinamento do modelo NER.
- *Patience*: O *patience* é um hiperparâmetro usado em combinação com a validação cruzada para determinar o momento de parar o treinamento. Ela especifica o número de épocas sem melhoria no desempenho de validação antes que o treinamento seja interrompido. O uso do *patience* ajuda a evitar o treinamento excessivo e reduz o risco de *overfitting* (PRECHELT, 1998).

2.1.2 Métricas de Avaliação e Matriz de Confusão

Ao desenvolver e treinar modelos de aprendizado de máquina, é essencial avaliar seu desempenho para determinar sua eficácia na tarefa específica. Existem várias métricas de avaliação que podem ser utilizadas para medir o desempenho de um modelo, dependendo da tarefa e dos objetivos do projeto.

A matriz de confusão é uma tabela que apresenta a quantidade de previsões corretas e incorretas de um modelo de aprendizado de máquina, organizadas por categoria.

Para problemas de classificação binária, a matriz de confusão consiste em quatro categorias:

1. Verdadeiros Positivos (VP): Previsões corretas da classe positiva.
2. Falsos Positivos (FP): Previsões incorretas da classe positiva.
3. Verdadeiros Negativos (VN): Previsões corretas da classe negativa.
4. Falsos Negativos (FN): Previsões incorretas da classe negativa.

A partir da matriz de confusão, várias métricas podem ser calculadas para avaliar o desempenho de um modelo de aprendizado de máquina:

- *Accuracy* (Acurácia): Representa a proporção de previsões corretas em relação ao total de previsões.

$$Acurácia = \frac{VP+VN}{VP+FP+VN+FN}$$

- *Precision* (Precisão): Indica a proporção de previsões corretas da classe positiva em relação ao total de previsões positivas e identifica a capacidade de evitar falsos positivos.

$$\text{Precisão} = \frac{VP}{VP+FP}$$

- *Recall* (Sensibilidade): Indica a capacidade do modelo de identificar corretamente as entidades de interesse e identifica a capacidade de evitar falsos negativos.

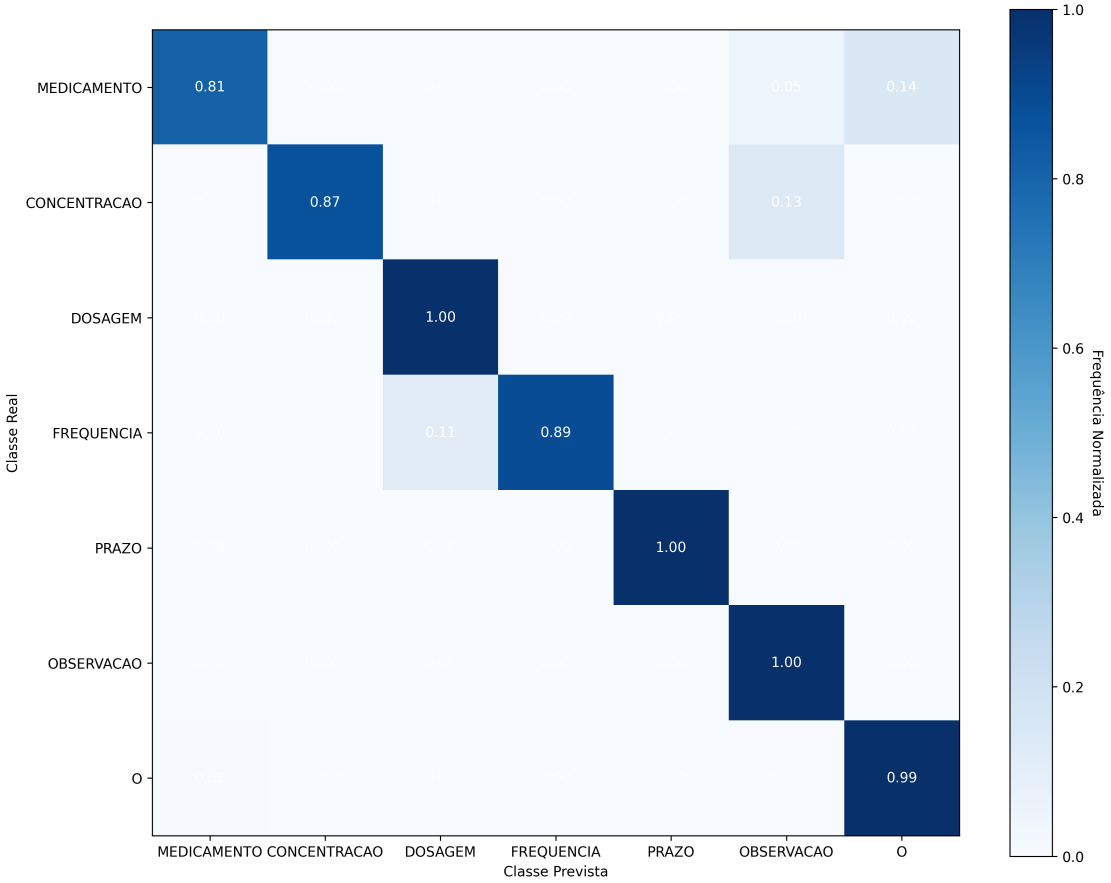
$$Recall = \frac{VP}{VP+FN}$$

- *F1-score*: Fornece uma avaliação geral do desempenho do modelo. Um valor de *F1-score* mais alto indica um equilíbrio adequado entre *recall* e precisão.

$$F1\text{-}score = 2 \times \frac{Precisao \times Recall}{Precisao + Recall}$$

Em problemas de classificação multiclasse, como o reconhecimento de entidades nomeadas, a matriz de confusão pode ser expandida para incluir todas as classes, e as métricas de avaliação podem ser calculadas para cada classe individualmente ou em média (SOKOLOVA; LAPALME, 2009). A Figura 2.2 apresenta um exemplo aplicado de uma classificação multiclasse:

Figura 2.2 – Exemplo de matriz de confusão multiclasse, com frequência normalizada



Fonte: O Autor

2.2 Processamento de Linguagem Natural

O processamento de linguagem natural (PLN) é uma área interdisciplinar da inteligência artificial que busca desenvolver algoritmos e técnicas capazes de analisar, entender e gerar texto em linguagem humana (JURAFSKY; MARTIN, 2019). O PLN tem sido aplicado em diversas tarefas, como análise de sentimentos, tradução automática e extração de informações. No contexto deste trabalho, o PLN é utilizado para identificar e extrair entidades relevantes de prescrições médicas eletrônicas.

2.2.1 Técnicas de Extração de Informação

A extração de informações é uma tarefa do PLN que visa identificar e extrair informações estruturadas de textos não estruturados (MOENS, 2006). Uma das técnicas mais

comuns de extração de informações é o reconhecimento de entidades nomeadas, que consiste em localizar e classificar entidades mencionadas no texto em categorias predefinidas, como pessoas, organizações e localizações (NADEAU; SEKINE, 2007).

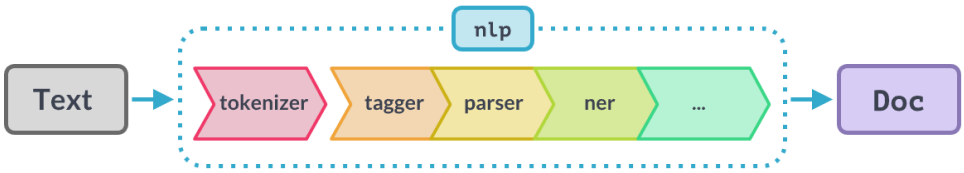
No contexto deste trabalho, a tarefa de reconhecimento de entidades nomeadas é adaptada para identificar e extrair entidades específicas das prescrições médicas, como medicamentos, concentrações, dosagens, frequências, prazos e observações. O modelo `pt_core_news_sm` do SpaCy, pré-treinado para o português, é utilizado como base para o treinamento do modelo NER.

2.2.2 SpaCy

SpaCy é uma biblioteca de PLN de código aberto amplamente utilizada para tarefas de processamento de texto em Python. Ele oferece modelos pré-treinados para várias línguas, incluindo o português, e suporta várias tarefas de PLN, como *tokenização*, lematização, análise morfológica, análise sintática e reconhecimento de entidades nomeadas (HONNIBAL; MONTANI, 2017).

O SpaCy organiza o processamento de texto em pipelines, que são sequências de componentes de processamento aplicados em ordem. Cada componente do pipeline é responsável por uma tarefa específica de PLN e pode modificar ou adicionar informações ao documento sendo processado, conforme a Figura 2.3:

Figura 2.3 – Pipelines de processamento de linguagem do SpaCy



Fonte: (SPACY, 2023)

Um pipeline típico do SpaCy pode incluir componentes como:

1. *Tokenização*: Divide o texto em unidades menores, como palavras e pontuação.
2. Análise morfológica: Identifica as características morfológicas das palavras, como gênero, número e tempo verbal.
3. Lematização: Converte as palavras para sua forma base ou raiz.
4. Análise sintática: Identifica a estrutura gramatical do texto e as relações entre as palavras.

5. NER: Como mencionado na subseção 2.2.1, localiza e classifica entidades mencionadas no texto em categorias predefinidas.

2.2.3 *Fine-Tuning*

O *fine-tuning* é uma técnica comum em aprendizado de máquina e aprendizado profundo que envolve o ajuste de um modelo pré-treinado em uma tarefa específica ou um conjunto de dados (YOSINSKI et al., 2014). Ao utilizar um modelo pré-treinado como ponto de partida, é possível aproveitar o conhecimento prévio aprendido pelo modelo em tarefas relacionadas e acelerar o processo de treinamento, resultando em um desempenho melhorado e menor necessidade de dados de treinamento em comparação com o treinamento do modelo do zero. Geralmente envolve duas etapas principais:

1. Inicialização do modelo: O modelo pré-treinado é carregado com os pesos e a arquitetura originais. Dependendo da tarefa, algumas camadas do modelo podem ser congeladas ou adaptadas, de modo que os pesos iniciais sejam mantidos ou modificados durante o treinamento.
2. Treinamento: O modelo é treinado no conjunto de dados específico da tarefa, utilizando uma taxa de aprendizado menor do que a taxa usada no treinamento original do modelo pré-treinado. Isso ajuda a preservar o conhecimento prévio aprendido pelo modelo e a ajustar os pesos do modelo de forma mais refinada para a tarefa específica.

No contexto deste trabalho, o *fine-tuning* é aplicado ao modelo NER do SpaCy. Os modelos `pt_core_news` do SpaCy variam em termos de tamanho e complexidade, o que pode afetar a precisão e o desempenho do modelo NER. Neste trabalho, optou-se em aplicar o *fine-tuning* no modelo `pt_core_news_sm`, pré-treinado para o português, devido ao seu equilíbrio entre precisão e eficiência computacional. Os pesos e a arquitetura do componente NER são ajustados para identificar e extrair entidades relacionadas às prescrições médicas, como medicamentos, concentrações, dosagens, frequências, prazos e observações.

Ao utilizar o *fine-tuning*, é possível aproveitar o conhecimento prévio do modelo `pt_core_news_sm` em relação à língua portuguesa e adaptá-lo para extrair informações específicas das prescrições médicas com maior precisão e menor necessidade de dados de treinamento.

2.3 Privacidade e ética em dados médicos

As pessoas envolvidas neste trabalho estão cientes da Lei Geral de Proteção de Dados Pessoais (LGPD), Lei nº 13.715/2018 (Presidência da República, 2018), principalmente pelo fato de dados relativos à saúde possuírem um alto grau de sensibilidade. Neste trabalho, entende-se que a prescrição médica é propriedade do paciente e este deverá consentir com um termo, autorizando a realizar da extração das informações.

Além disso, entende-se que as prescrições médicas eletrônicas podem contem informações sensíveis de terceiros. Estes dados serão anonimizados ou retirados durante o processo, garantindo a privacidade das mesmas.

3 TRABALHOS RELACIONADOS E ANÁLISE DE MERCADO

3.1 Trabalhos Relacionados

A extração de informações em prescrições médicas é um tópico de pesquisa importante, pois possibilita a análise e o monitoramento do uso de medicamentos e a identificação de eventos adversos. Vários trabalhos foram realizados para desenvolver métodos eficientes de extração de informações, utilizando técnicas de Processamento de Linguagem Natural (PLN) e aprendizado de máquina.

3.1.1 *Neural Architectures for Named Entity Recognition*

Neste artigo, Lample et al. (2016) apresenta as arquiteturas neurais para reconhecimento de entidades nomeadas. Essas arquiteturas são baseadas em redes neurais recorrentes (RNNs) com redes de memória de longo prazo (LSTMs) para capturar informações contextuais do texto.

A arquitetura proposta por Lample et al. (2016) consiste em quatro componentes principais:

1. Representação de palavras: As palavras no texto são representadas por vetores contínuos (*word embeddings*) que capturam informações semânticas e sintáticas. Esses *embeddings* são pré-treinados em grandes conjuntos de dados e ajustados durante o treinamento do modelo NER.
2. Representação de caracteres: Além das *word embeddings*, a arquitetura também utiliza *embeddings* de caracteres para representar as palavras. Isso ajuda o modelo a lidar com palavras desconhecidas e morfologia complexa. As representações de caracteres são obtidas através de uma rede neural convolucional (CNN) aplicada a sequências de caracteres das palavras.
3. RNNs e LSTMs: As representações de palavras e caracteres são combinadas e alimentadas em uma rede neural recorrente com células LSTM. A RNN captura informações contextuais do texto, permitindo que o modelo identifique entidades com base no contexto em que aparecem.
4. Decodificação: A saída da RNN é processada por uma camada de decodificação para gerar as entidade para cada palavra. Lample et al. (2016) utilizam uma téc-

nica de decodificação chamada CRF (*Conditional Random Fields*) para modelar dependências entre etiquetas adjacentes e melhorar a precisão do modelo NER.

No contexto deste trabalho, este artigo fornece uma base teórica para o desenvolvimento de modelos NER eficientes. No entanto, existem diferenças e limitações quando comparado a este trabalho.

O artigo de Lample et al. (2016) não se concentra especificamente na extração de informações de prescrições médicas. Portanto, os modelos NER apresentados no artigo não são imediatamente aplicáveis ao domínio deste trabalho sem adaptações ou ajustes.

Ainda, neste trabalho foi realizado um pré-processamento das amostras de prescrições médicas e um pós-processamento utilizando expressões regulares para eliminar possíveis resquícios nas entidades detectadas e realizar padronizações. O artigo de Lample et al. (2016) não aborda explicitamente esses aspectos, o que também limita a aplicabilidade imediata dos modelos neste contexto.

3.1.2 *CADEC: A Corpus of Adverse Drug Event Annotations*

Karimi, Metke-Jimenez and Kemp (2015) introduziram o corpus CADEC (Corpus of Adverse Drug Event Annotations), uma coleção de textos médicos anotados com informações sobre eventos adversos a medicamentos. O corpus foi desenvolvido com o objetivo de fornecer um recurso para treinar e avaliar modelos de PLN na tarefa de extrair informações relacionadas a eventos adversos em textos médicos.

O CADEC contém anotações de eventos adversos coletadas a partir de fóruns online onde pacientes compartilham suas experiências com medicamentos. As anotações incluem informações sobre medicamentos, sintomas, doenças, procedimentos médicos, entre outros.

O desenvolvimento do corpus CADEC foi importante para impulsionar pesquisas na área de PLN aplicada a eventos adversos a medicamentos. A disponibilidade de um conjunto de dados anotado permite que pesquisadores treinem e avaliem modelos de aprendizado de máquina e PLN com foco na identificação e extração de informações relevantes sobre eventos adversos em textos médicos.

Apesar da relação na área de aplicação com este trabalho, o CADEC não é diretamente aplicável neste contexto, pois seu foco está em eventos adversos a medicamentos e as anotações são provenientes de postagens de fóruns de pacientes. O contexto e a

linguagem usados em prescrições médicas e postagens de fóruns de pacientes são bem diferentes, o que pode resultar em uma performance bem abaixo do esperado se o CADEC fosse utilizado para treinar o modelo NER.

3.1.3 *Introduction to the CoNLL-2003 Shared Task*

Outro trabalho relacionado é o de Sang and Meulder (2003), que apresentam o CoNLL-2003, uma tarefa compartilhada para o reconhecimento de entidades nomeadas independente de idioma. A tarefa teve como objetivo estimular o desenvolvimento e a avaliação de algoritmos de reconhecimento de entidades nomeadas que possam ser aplicados a diferentes idiomas, incentivando a colaboração entre pesquisadores e o compartilhamento de recursos e ideias.

No CoNLL-2003, os participantes foram desafiados a desenvolver sistemas de reconhecimento de entidades nomeadas para dois idiomas, inglês e alemão. Os sistemas deveriam identificar e classificar quatro tipos de entidades nomeadas: pessoas, organizações, localizações e nomes de eventos ou produtos. Para cada idioma, os organizadores forneceram um conjunto de dados anotado, dividido em conjuntos de treinamento, desenvolvimento e teste.

O CoNLL-2003 teve um impacto significativo no campo de reconhecimento de entidades nomeadas, pois proporcionou um conjunto de dados padronizado e uma métrica de avaliação comum para comparar diferentes abordagens e algoritmos. Além disso, o desafio estimulou o desenvolvimento de métodos de reconhecimento de entidades nomeadas mais robustos e eficientes, capazes de lidar com variações linguísticas e adaptáveis a diferentes idiomas.

O SpaCy, biblioteca utilizada neste trabalho para treinar o modelo NER, utiliza várias fontes para treinar seus modelos, incluindo o CoNLL-2003. O modelo NER em português do SpaCy não é treinado especificamente no conjunto de dados CoNLL-2003, já que o CoNLL-2003 abrange apenas inglês e alemão, mas foram utilizados conjuntos de dados semelhantes ou adaptados.

Ao adaptar o modelo NER do SpaCy para o contexto de prescrições médicas, é possível aproveitar os avanços no campo de reconhecimento de entidades nomeadas, muitos dos quais foram impulsionados por iniciativas como o CoNLL-2003, para obter melhores resultados na extração de informações relevantes das prescrições médicas.

3.2 Análise de Mercado

Nesta seção, será apresentada uma análise comparativa de diferentes ferramentas e serviços de extração de informações de documentos médicos, focando no reconhecimento de entidades nomeadas. Essa comparação ajudará a contextualizar a abordagem proposta neste trabalho em relação a soluções existentes no campo do processamento de linguagem natural aplicado ao domínio médico.

Como um serviço de PLN, a Amazon Comprehend Medical utiliza *machine learning* para extrair informações médicas de texto não estruturado. Ele oferece suporte à extração de diversas entidades. No entanto, ele não suporta treinamento personalizado ou *fine-tuning*, o que pode limitar sua aplicabilidade em algumas situações (Amazon Web Services, Inc., 2022).

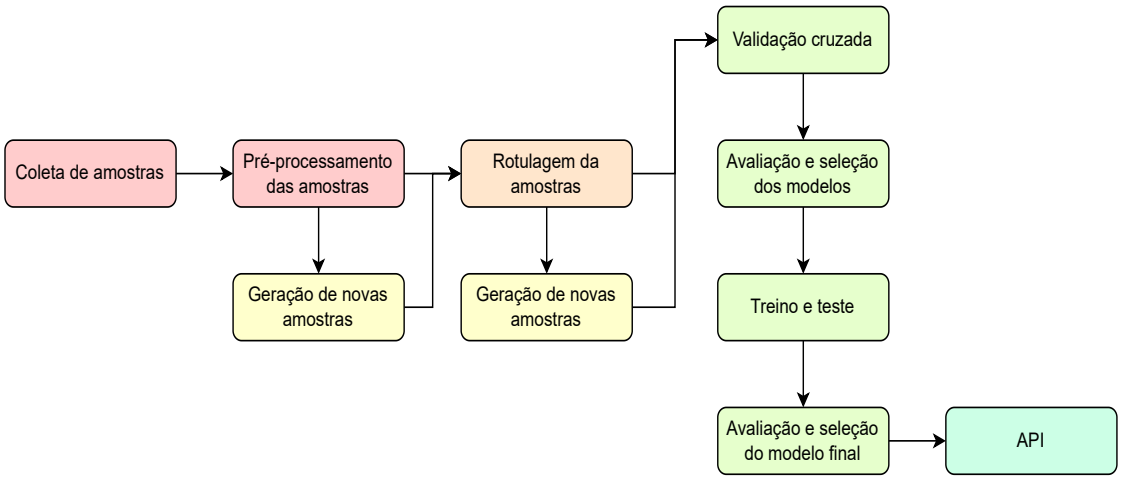
MedaCy é uma biblioteca Python de código aberto para PLN no domínio médico. Ela permite treinamento personalizado e reconhecimento de entidades nomeadas com base em modelos pré-treinados. A flexibilidade de treinar modelos personalizados pode ser uma vantagem em relação ao Amazon Comprehend Medical, mas pode exigir mais esforço e tempo de desenvolvimento (NEVES; TAGGER, 2022).

CLAMP (*Clinical Language Annotation, Modeling, and Processing Toolkit*) é um conjunto de ferramentas de processamento de linguagem natural desenvolvido especificamente para o domínio médico. Ele oferece uma interface gráfica do usuário e permite a criação de pipelines personalizados para reconhecimento de entidades nomeadas, normalização de conceitos e outras tarefas relacionadas ao PLN. A ferramenta inclui componentes pré-treinados e pode ser ajustada para extrair informações específicas de documentos médicos (SOYSAL et al., 2018). Porém, o CLAMP é uma ferramenta mais geral para trabalhar com textos clínicos e biomédicos, enquanto este trabalho se concentra especificamente em prescrições médicas. Por se tratar de uma biblioteca de PLN de propósito geral, o SpaCy demonstrou ser mais fácil de adaptar às necessidades específicas, incluindo também a maior facilidade de integração com a API desenvolvida neste trabalho.

4 METODOLOGIA

O processo metodológico foi dividido em cinco etapas principais, cada uma das quais será abordada em detalhes nas seções a seguir. O fluxograma geral da metodologia aplicada pode ser visto na Figura 4.1:

Figura 4.1 – Fluxograma da metodologia aplicada



Fonte: O Autor

4.1 Coleta e pré-processamento das amostras

A coleta e o pré-processamento das amostras são etapas cruciais para garantir a qualidade e a eficácia do modelo de extração de entidades desenvolvido. Nesta seção, será descrito o processo de obtenção das amostras de prescrições médicas e a abordagem adotada para pré-processá-las.

4.1.1 Coleta das amostras

As amostras de prescrições médicas eletrônicas foram coletadas de maneira informal, por meio de conhecidos dos autor. No total, obteve-se 34 prescrições médicas em formato PDF, sendo 27 originárias do Conselho Regional de Medicina do Estado do Rio Grande do Sul (CREMERS), cinco da plataforma Memed e duas da Unimed. Das 34 prescrições médicas coletadas, 7 amostras foram descartadas por conter alguma irregularidade no formato do arquivo, restando 27 amostras.

4.1.2 Pré-processamento das amostras

Para extrair informações relevantes das prescrições médicas coletadas, realizou-se um pré-processamento através de *scripts* em Python, que consistiu nas seguintes etapas:

1. Conversão dos arquivos PDF para HTML: Primeiramente, os documentos em PDF foram convertidos para o formato HTML. Os arquivos PDF foram processados com a ajuda das classes `pdfResourceManager`, `pdfPageInterpreter`, `htmlConverter`, `LAParams` e `pdfPage` da biblioteca pdfminer[1]. A conversão foi realizada para facilitar a extração de texto e a manipulação dos dados.
2. Extração de textos relevantes: Com os arquivos em formato HTML, procedeu-se à extração das informações relevantes para o treinamento do modelo. Utilizando a biblioteca BeautifulSoup[2], foram identificados e extraídos os elementos de interesse, a partir das *tags* e atributos dos arquivos HTML, definindo limites para evitar capturar informações pessoais do paciente e do médico, e capturar apenas as informações relevantes, como medicamento, concentração, dosagem, frequência, prazo e observação. Os elementos selecionados tiveram seus textos extraídos, e as strings resultantes foram processadas para remover acentuação e traços, além de realizar a *tokenização* de quebras de linhas e espaços tipográficos. Para exemplificar este processo, um exemplo de prescrição médica eletrônica em PDF pode ser visto em A.1 e o HTML convertido em A.2 Abaixo, segue o exemplo do texto extraído, levemente adaptado para fins de visualização:

Figura 4.2 – Texto extraído do HTML representado na Figura A.2

```
Uso Interno: {newline} Amoxicilina + acido clavulanico 400mg / 5ml ..............
1 vidro (140ml) {newline} Administrar 5 ml (cinco) de 12/12 horas por 10 dias.
{newline} ### Guardar na geladeira {newline} Bifilac Geflora
saches .....................1 caixa {newline}  Tomar 1 sache 1x dia , por 10 dias =
no intervalo dos horarios de antibiotico almoco. {newline} Diluir em agua ou suco
e administrar imediatamente. {newline}
```

Fonte: O Autor

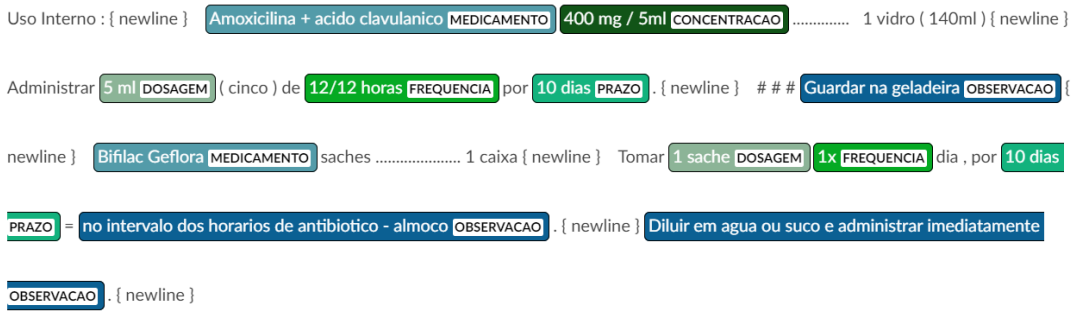
Após a conclusão do pré-processamento, obteve-se um conjunto de dados contendo textos relevantes das prescrições médicas, pronto para ser utilizado nas etapas subsequentes de rotulagem, geração de amostras e treinamento do modelo NER.

[1] https://pypi.org/project/pdfminer/
[2] https://pypi.org/project/beautifulsoup4/

4.2 Rotulagem das amostras

A rotulagem dos dados é uma etapa fundamental para a criação de um modelo de aprendizado supervisionado, como o modelo NER desenvolvido. Utilizou-se a ferramenta UBIAI[3] para rotular as amostras pré-processadas, identificando entidades como medicamento, concentração, dosagem, frequência, prazo e observação, conforme Figura 4.3.

Figura 4.3 – Exemplo de rotulagem das amostras com a ferramenta UBIAI



Fonte: O Autor

É importante observar que nem todas as amostras continham todas as entidades. Depois de rotuladas, as amostras foram exportadas no formato JSON, com o mapeamento de cada entidade rotulada, indicando a posição do caractere de início e o de fim do texto rotulado, conforme Figura 4.4 abaixo:

Figura 4.4 – Exemplo de arquivo JSON exportado pela ferramenta UBIAI após rotulagem das amostras


```
"documentName": "03_receita_cremers.txt",

"document":
   "Uso Interno: {newline}   Amoxicilina + acido clavulanico 400mg / 5ml .............  1 vidro (140ml) {newline}
Administrar 5 ml (cinco) de 12/12 horas por 10 dias. {newline}   ### Guardar na geladeira {newline}   Bifilac Geflora
saches .....................1 caixa {newline}   Tomar 1 sache 1x dia , por 10 dias = no intervalo dos horarios de
antibiotico - almoco. {newline} Diluir em agua ou suco e administrar imediatamente. {newline} ",

"annotation":
   [{"start": 57, "end": 68, "label": "CONCENTRACAO", "text": "400mg / 5ml", "propertiesList": [], "commentsList": []},
   {"start": 125, "end": 129, "label": "DOSAGEM", "text": "5 ml", "propertiesList": [], "commentsList": []},
   {"start": 141, "end": 152, "label": "FREQUENCIA", "text": "12/12 horas", "propertiesList": [], "commentsList": []}]
```


Fonte: O Autor

4.3 Geração de amostras

Para aumentar a quantidade de dados disponíveis e garantir a robustez do modelo, foram gerados dois grupos adicionais de amostras, utilizando diferentes estratégias.

[3] https://ubiai.tools/

4.3.1 Grupo de amostras gerado a partir de dados originais

Neste grupo, o ponto de partida foram as 27 amostras originais pré-processadas. Criou-se manualmente uma amostra adicional para cada uma amostra original, alterando levemente a estrutura do texto e incluindo, alterando ou removendo informações, de forma que a amostra gerada fosse plausível com a realidade. Por exemplo, a amostra gerada a partir da amostra representada na Figura 4.2 pode ser vista na Figura 4.5:

Figura 4.5 – Amostra gerada a partir da amostra representada na Figura 4.2

```
Uso Interno: {newline} - Dipirona 1g ..............  140ml {newline}   Administrar
8 ml (cinco) de 4/4 horas por 5 dias. {newline}  Aguardar 30 minutos após a
administração para deitar {newline} - Maleato de
dexclorfeniramina ..................... 1 caixa {newline}    Tomar 4 sache 1x
dia , por 10 dias = não ingerir bebidas com cafeína durante o tratamento {newline}
Diluir em agua ou suco e administrar imediatamente. {newline}
```

Fonte: O Autor

4.3.2 Grupo de amostras gerado a partir da variação de dados originais

Para gerar o segundo grupo de amostras geradas, foram obtidos aproximadamente 19 mil nomes distintos registrados como medicamentos na Agência Nacional de Vigilância Sanitária (Anvisa) (ANVISA, 2023a), incluindo genérico e vacinas, bem como exemplos de concentrações de medicamentos que contenham um único insumo farmacêutico ativo (ANVISA, 2023b) e exemplos de concentrações de medicamentos que contenham dois ou mais insumos farmacêuticos ativos em uma única forma farmacêutica (ANVISA, 2023c). Além disso, criou-se manualmente variações de dosagens, frequências, prazos e observações, contendo inclusive erros de escrita.

Com o objetivo de gerar aproximadamente 30 mil amostras, este grupo partiu do primeiro grupo de amostras geradas, já rotuladas e exportadas no formato JSON. Desta forma, para automatizar este processo, foi desenvolvido um *script* em Python capaz de selecionar aleatoriamente amostras do primeiro grupo, e substituir o texto principal e os textos referentes às entidades rotuladas originalmente por exemplos obtidos e criados manualmente, com a restrição de serem entidades equivalentes. As posições dos caracteres de início e fim também foram atualizados. Após a geração das amostras, as novas anotações foram salvas no formato JSON, seguindo o mesmo formato dos dados rotulados previamente.

Com a geração de amostras adicionais e a rotulagem dos dados, é garantido que

o modelo NER seja treinado com um conjunto de dados diversificado e representativo, aumentando sua capacidade de generalização e melhorando a qualidade das predições em casos reais, mesmo possuindo um desbalanceamento entre as classes.

4.4 Desenvolvimento da API

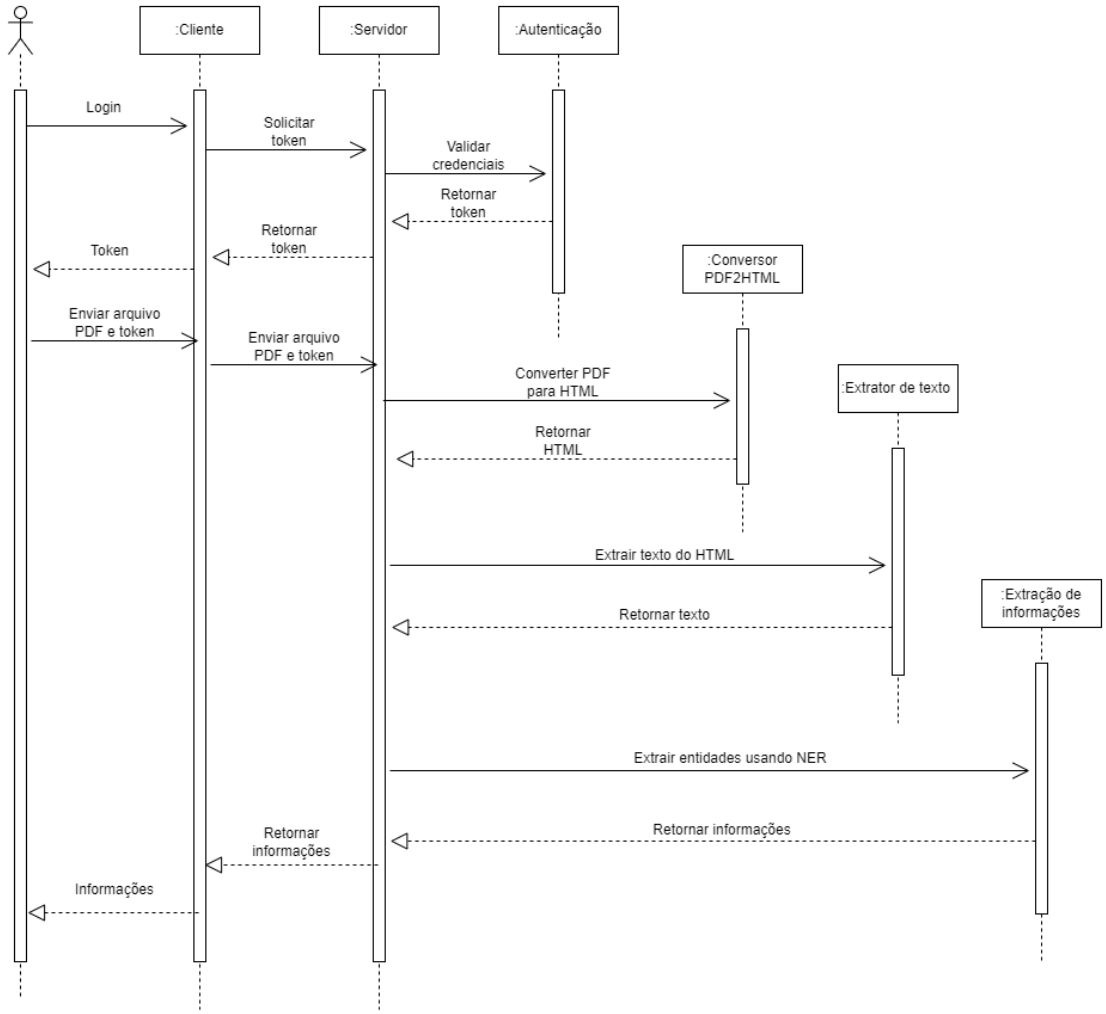
A fim de fornecer uma interface eficiente para interagir com o modelo NER treinado e realizar a extração de entidades a partir de prescrições médicas, foi desenvolvida uma API utilizando FastAPI, um moderno *framework* Python para construção de APIs que possui alta performance e fácil implementação.

A arquitetura da API é composta por diversos componentes, incluindo a autenticação de usuários, o processamento de arquivos PDF e a utilização do modelo NER para extração das entidades. Além disso, a API realiza o pós-processamento das entidades detectadas utilizando expressões regulares e regras lógicas para padronizar e validar os resultados retornados. O processo de extração de informações ocorre conforme representado na Figura 4.6.

A conversão de PDF para HTML e a extração de texto ocorrem exatamente nos mesmos moldes que o pré-processamento descrito na subseção 4.1.2, garantindo a compatibilidade entre o texto recebido e as amostras usadas para treinamento.

Para garantir a segurança e a integridade dos dados, a API utiliza autenticação baseada em token JWT (JSON Web Token). Os usuários devem fornecer suas credenciais para obter um *token* de acesso, que será utilizado para autorizar o acesso aos *endpoints* da API.

Figura 4.6 – Diagrama de sequência da API desenvolvida



Fonte: O Autor

Em resumo, a API desenvolvida oferece uma solução completa para a extração de informações de prescrições médicas, permitindo que aplicações externas possam se beneficiar do modelo NER treinado e das técnicas de pós-processamento para obter resultados precisos e padronizados.

5 EXECUÇÃO DO *PIPELINE*

Neste capítulo, será detalhado o processo de treinamento do modelo NER com a biblioteca SpaCy (versão 3.5.1), realizado em uma máquina com as seguintes configurações:

- Processador: AMD Ryzen™ 5 3600X.
- RAM: 16GB.
- SSD: 512 GB.
- SO: Subsistema do Windows para Linux (Ubuntu 20.04).
- GPU: Radeon™ RX 580 (não utilizada no treinamento).

5.1 Configuração do modelo

Primeiramente, obteve-se o modelo `pt_core_news_sm` compatível com a versão do SpaCy utilizada. Esse modelo contém diversos pipelines. No entanto, para o propósito específico deste trabalho, focou-se no reconhecimento de entidades nomeadas e, portanto, foi decidido manter apenas o pipeline NER, removendo o restante dos pipelines.

Ao manter apenas o pipeline NER, garantimos que o modelo se concentre na tarefa de identificar e classificar as entidades de interesse. Dessa forma, foram adicionadas as entidades ao pipeline NER: **MEDICAMENTO**, **CONCENTRACAO**, **DOSAGEM**, **FREQUENCIA**, **PRAZO** e **OBSERVACAO**. Essa abordagem simplificada permite a otimização do modelo para a tarefa específica de extração de informações relevantes de prescrições médicas, resultando em melhor desempenho e maior precisão.

5.2 Preparação dos dados

Para garantir a qualidade e a consistência dos dados rotulados, foi utilizada a função `offsets_to_biluo_tags`, do SpaCy, para validá-los. Essa função converte as anotações de entidades nos dados de treinamento para o formato BILUO, que é amplamente utilizado em tarefas de NER.

O formato BILUO é uma notação específica que representa a posição das entida-

des no texto. As seguintes etiquetas são usadas:

- B (Begin): indica o início de uma entidade de várias palavras.
- I (Inside): indica uma palavra que está no meio de uma entidade de várias palavras.
- L (Last): indica a última palavra de uma entidade de várias palavras.
- U (Unit): indica uma entidade de uma única palavra.
- O (Outside): indica uma palavra que não faz parte de uma entidade.

A utilização da função `offsets_to_biluo_tags` permite identificar e corrigir possíveis inconsistências nas anotações de entidades, garantindo a qualidade dos dados de treinamento e, consequentemente, a eficácia do modelo NER treinado. Com os dados adequadamente preparados e validados, é possível seguir com o processo de treinamento e avaliação do modelo NER do SpaCy.

5.3 Validação cruzada

Para entender o comportamento dos modelos em relação aos dados e hiperparâmetros, foi realizada a validação cruzada. De forma a ficar claro, iremos adotar novas nomenclaturas para os grupos de amostras:

- Amostras originais pré-processadas: serão chamadas de grupo de amostras 1.
- Grupo de amostras gerado a partir de dados originais: será chamada de grupo de amostras 2.
- Grupo de amostras gerado a partir da variação de dados originais: será chamada de grupo de amostras 3.

Assim, para cada grupo de amostra, utilizou-se diversas combinações de hiperparâmetros, gerando inúmeros modelos. Abaixo, seguem os parâmetros e hiperparâmetros utilizados:

- *Folds*: foi determinado o número de 10 *folds* com dados estratificados. Os dados estratificados garantem que cada *fold* terá a mesma representatividade de dados de cada classe, exceto se a quantidade de dados em alguma classe for menor que 10.
- Otimizador: Adam.
- Taxa de aprendizado: 0.001.
- Épocas: Limite máximo de 200 épocas.

- *Patience*: interrompe o treinamento caso não haja melhoria no desempenho após cinco épocas consecutivas.
- Hiperparâmetros variados:

  Dropout: foram testados três valores distintos, 0.0, 0.2 e 0.5.

  Batch size: também foram testados três valores distintos, 4, 32 e 64.

O objetivo de variar os hiperparâmetros é encontrar a combinação ideal que resulta no melhor desempenho do modelo. Ao ajustar os hiperparâmetros, como *dropout* e *batch size*, é possível explorar diferentes configurações de treinamento, buscando uma solução que minimize a função de custo de entropia cruzada e maximize a precisão, sem comprometer a capacidade do modelo de generalizar para novos dados, evitando *overfitting* ou *underfitting*.

No caso específico do *dropout*, a variação desse hiperparâmetro tem como objetivo encontrar o equilíbrio adequado na regularização do modelo, visto que as amostras do trabalho possuem estruturas muito semelhantes, reduzindo a dependência de neurônios específicos e evitando *overfitting*.

Quanto ao *batch size*, testar diferentes valores visa otimizar a eficiência computacional e a convergência do treinamento.

Após a conclusão do treinamento, os modelos com os menores valores de perda (*loss*), baseado em entropia cruzada, foram armazenados. Estes modelos serão utilizados para avaliar o desempenho geral e individual de cada *fold* no conjunto de dados.

5.4 Avaliação dos modelos

A fim de avaliar o desempenho dos modelos, empregou-se duas abordagens complementares: a matriz de confusão e o cálculo de métricas *F1-score*, *recall* e precisão.

A matriz de confusão permite visualizar a relação entre as previsões do modelo e os valores reais dos dados de validação, desta forma, possibilitando avaliar o reconhecimento das entidades. A partir da biblioteca scikit-learn, construiu-se uma matriz de confusão para cada *fold*, facilitando a identificação de possíveis padrões e áreas de melhoria no desempenho do modelo.

Em paralelo, foram calculadas as métricas *F1-score*, *recall* e precisão diretamente pelo SpaCy, utilizando a função `evaluate()`.

5.5 Treinamento final dos modelos selecionados

Após a avaliação dos modelos utilizando a validação cruzada, selecionou-se os melhores modelos com base nas métricas de desempenho obtidas. Nesta etapa, realizou-se o *fine-tuning* desses modelos selecionados utilizando todas as amostras disponíveis dentro dos grupos de amostras correspondentes. O objetivo dessa etapa é refinar ainda mais os modelos e maximizar seu desempenho ao utilizar o conjunto completo de dados de treinamento.

Para garantir uma avaliação justa e comparativa entre os modelos, foi aplicado a cada um deles a um conjunto de teste uniforme. Esse conjunto de teste não foi utilizado durante o treinamento ou a validação cruzada e serve para fornecer uma estimativa realista do desempenho dos modelos em dados não vistos.

Nesta etapa, também foram geradas métricas adicionais e a matriz de confusão para cada modelo, utilizando o conjunto de teste mencionado anteriormente. Essa análise ajuda a identificar áreas onde os modelos podem estar tendo dificuldades e fornece ideias sobre possíveis melhorias e ajustes futuros.

Com base nas métricas e na matriz de confusão, é possível comparar os modelos e selecionar aquele que apresenta o melhor desempenho geral. Esse modelo selecionado servirá como a base para extrair informações relevantes de prescrições médicas em aplicações práticas, possibilitando a construção de soluções eficientes e precisas para o processamento desses documentos.

6 RESULTADOS

Neste capítulo, o objetivo é avaliar a eficácia da ferramenta proposta, utilizando técnicas de processamento de linguagem natural e aprendizado de máquina. Para isso, serão analisados os resultados obtidos no treinamento do modelo NER, tanto na validação cruzada quanto no treinamento final, bem como os resultados dos testes realizados com a API desenvolvida.

6.1 Avaliação dos modelos de validação cruzada

Inicialmente, foi realizada a validação cruzada com 27 modelos NER treinados utilizando o SpaCy. Essa etapa foi crucial para avaliar o desempenho dos modelos em diferentes cenários, permitindo a seleção dos melhores modelos para uma segunda fase de treinamento e avaliação. Nessa segunda fase, todas as amostras disponíveis nos grupos dos modelos selecionados foram utilizadas, garantindo que o modelo escolhido para a API fosse o mais adequado para os objetivos da ferramenta. Para cada modelo, foram calculadas métricas como *F1-score*, *recall* e precisão, bem como a matriz de confusão, a fim de verificar o desempenho em cada classe de entidade (medicamento, concentração, dosagem, frequência, prazo e observação).

Conforme descrito na subseção 5.3, optou-se por variar os hiperparâmetros de *dropout* e *batch size*, aplicando-os a cada grupo de amostras:

- *Dropout*: 0.0, 0.2 e 0.5.
- *Batch size*: 4, 32 e 64.
- Grupos de amostras: 1, 2 e 3.

As configurações completas do modelos podem ser visualizadas na Tabela B.1.

A validação cruzada foi realizada com 10 *folds* estratificados. A estratificação buscou equilibrar a representação de cada entidade em cada *fold*, considerando que nem todas as amostras possuem todas as entidades.

A avaliação foi conduzida de três maneiras:

1. Desempenho em relação ao grupo de amostras: Análise do desempenho dos modelos conforme o aumento do número de amostras.
2. Desempenho em relação ao *dropout*: Análise do desempenho dos modelos para

diferentes valores de *dropout*.

3. Desempenho em relação ao *batch size*: Análise do desempenho dos modelos para diferentes valores de *batch size*.

6.1.1 Desempenho em relação ao grupo de amostras

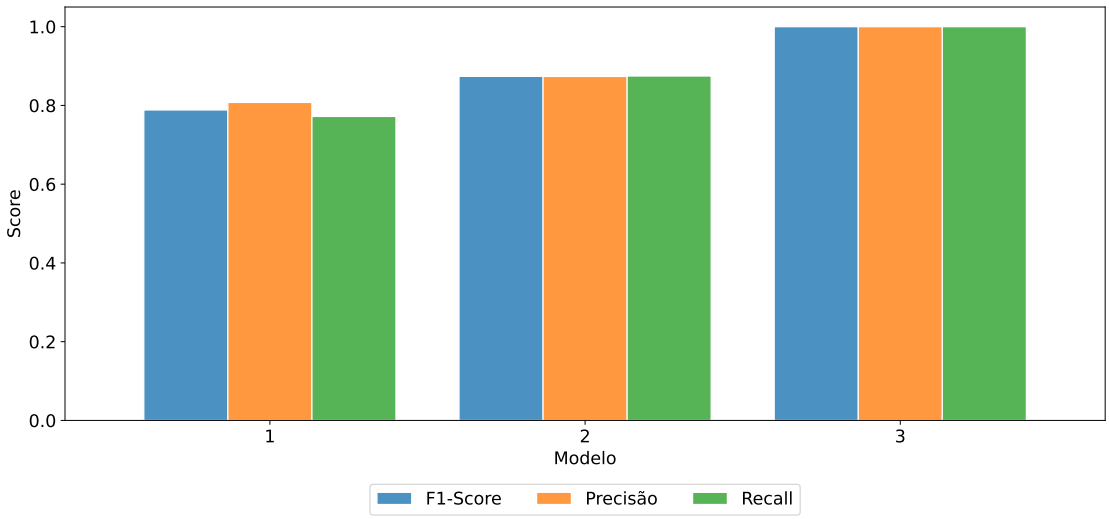
Para essa avaliação, foram selecionados os modelos 1, 2 e 3, pois não usam *dropout* durante o treinamento e possuem o menor *batch size*. Essas duas condições foram colocadas com o objetivo de diminuir ao máximo a interferência destes hiperparâmetros na avaliação do desempenho em relação ao grupo de amostra. Ao comparar esses três modelos, o objetivo é verificar se há uma melhora de desempenho ao utilizar mais dados, mesmo que artificiais.

Tabela 6.1 – Modelos selecionados para a avaliação do desempenho em relação ao grupo de amostras

Modelo	*Dropout*	*Batch size*	Grupo
1	0.0	4	1
2	0.0	4	2
3	0.0	4	3

Fonte: O Autor

Figura 6.1 – Métricas dos Modelos 1, 2 e 3

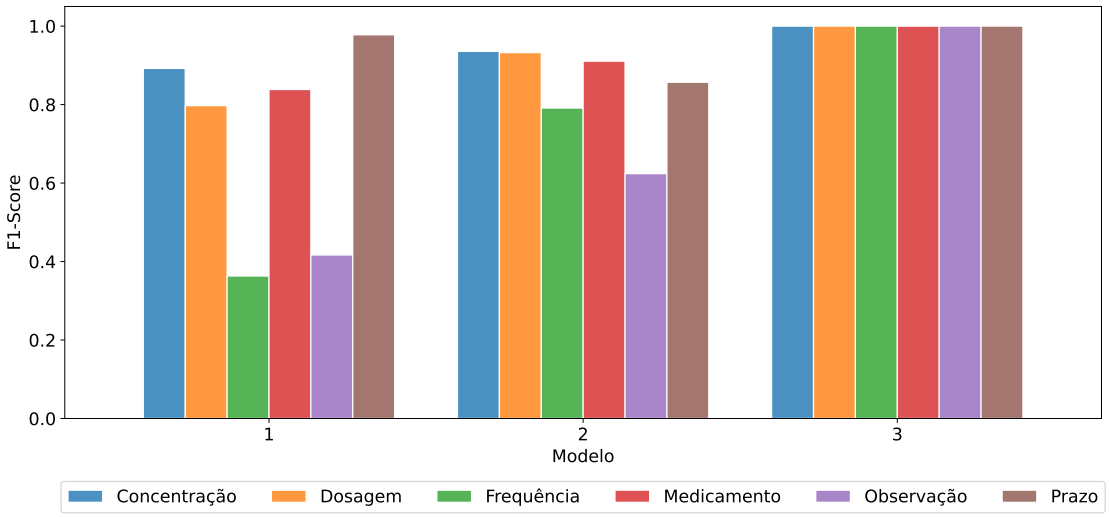


Fonte: O Autor

Ao detalhar esses modelos por entidades, conforme a Figura 6.2, é possível analisar claramente que entidades como frequência e observação, que possuíam um desem-

penho ruim nas amostras originais, obtiveram melhoria de desempenho. O desempenho insatisfatório dessas entidades no modelo 1 é compreensível, visto que são entidades que não aparecem necessariamente nas prescrições médicas e, somado ao fato de ser um conjunto de amostras pequeno. Ao expandir o conjunto de amostras, conforme explicado na seção 4.3, houve um aumento leve no desempenho de entidades que aparecem obrigatoriamente nas prescrições e de maneira expressiva no desempenho de entidades menos frequentes.

Figura 6.2 – F1-Score das entidades dos modelos 1, 2 e 3



Fonte: O Autor

6.1.2 Desempenho em relação ao *dropout*

Em contraste com a avaliação anterior, o objetivo desta análise é comparar o desempenho dos modelos para diferentes valores de *dropout*. Serão avaliados todos os modelos com *batch size* igual a 4, agrupados por grupos de amostras, eliminando a interferência da quantidade de amostras e *batch size* na comparação. A seguir estão os modelos selecionados:

- Grupo de amostras 1: foram selecionados os modelos 1, 10 e 19.
- Grupo de amostras 2: foram selecionados os modelos 2, 11 e 20.
- Grupo de amostras 3: foram selecionados os modelos 3, 12 e 21.

Os modelos serão analisados por grupos. Assim, primeiramente serão analisados os modelos treinados do grupo 1, em seguida serão analisados os modelos treinados do grupo 2, e por último os modelos treinados do grupo 3.

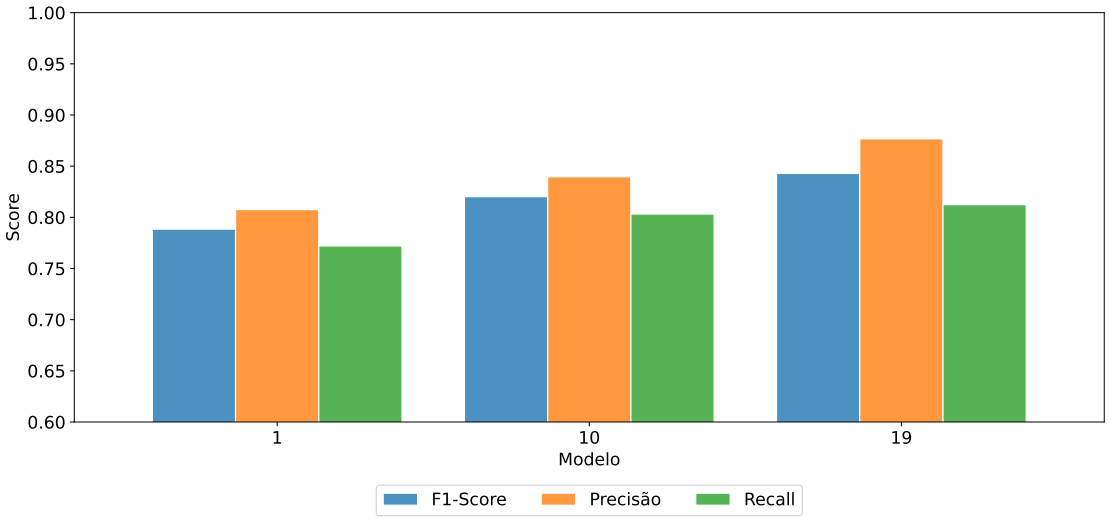
Tabela 6.2 – Modelos selecionados para a avaliação do desempenho em relação ao *dropout*

Modelo	*Dropout*	*Batch size*	Grupo
1	0.0	4	1
2	0.0	4	2
3	0.0	4	3
10	0.2	4	1
11	0.2	4	2
12	0.2	4	3
19	0.5	4	1
20	0.5	4	2
21	0.5	4	3

Fonte: O Autor

Conforme mostrado na Figura 6.3, entre os modelos selecionados do grupo de amostras 1, o modelo 19 possui melhor precisão e *recall*. Este modelo possui o maior valor de *dropout* entre os analisados. Entende-se que o *dropout* pode ser mais eficaz em conjuntos de amostras menores porque esses conjuntos são mais propensos ao *overfitting*. Quando um conjunto de dados é pequeno, o modelo tem mais dificuldade em aprender padrões generalizáveis e é mais provável que se ajuste demais aos ruídos presentes nos dados.

Figura 6.3 – Métricas dos modelos 1, 10 e 19. A escala do eixo *Score* se inicia em 0.6

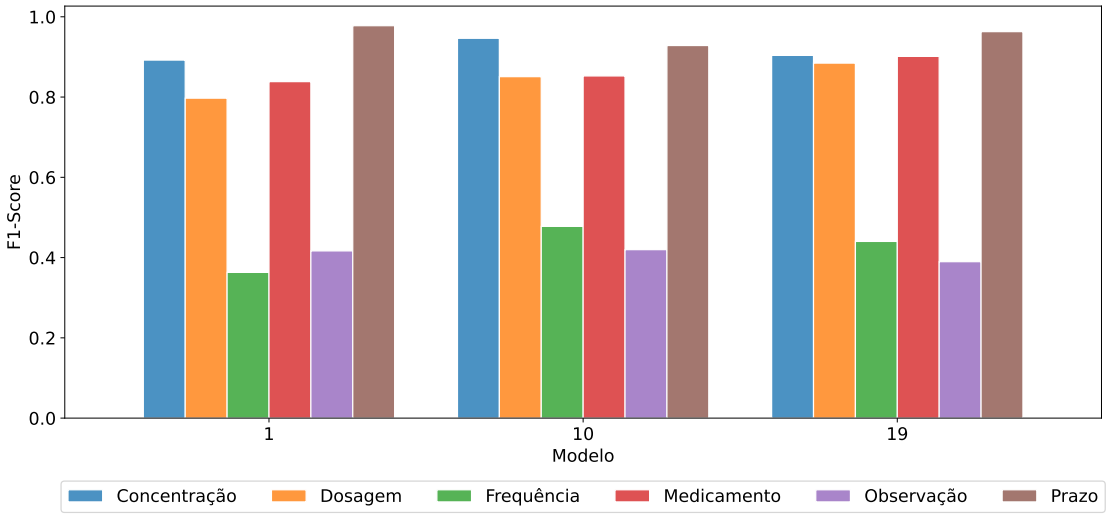


Fonte: O Autor

Na Figura 6.4, os três modelos são detalhados por entidade. As entidades do modelo 19, que se destacaram inicialmente, estão mais alinhadas do que as entidades dos modelos 1 e 10, exceto pela **FREQUENCIA** e **OBSERVACAO**, conforme esperado. Estas duas entidades possuem menos amostras, e ainda, no caso da **OBSERVACAO**, se

trata de uma entidade de difícil reconhecimento. A harmonia entre as outras entidades, analisada juntamente ao desempenho médio do modelo, indica um melhor desempenho geral em comparação aos outros modelos.

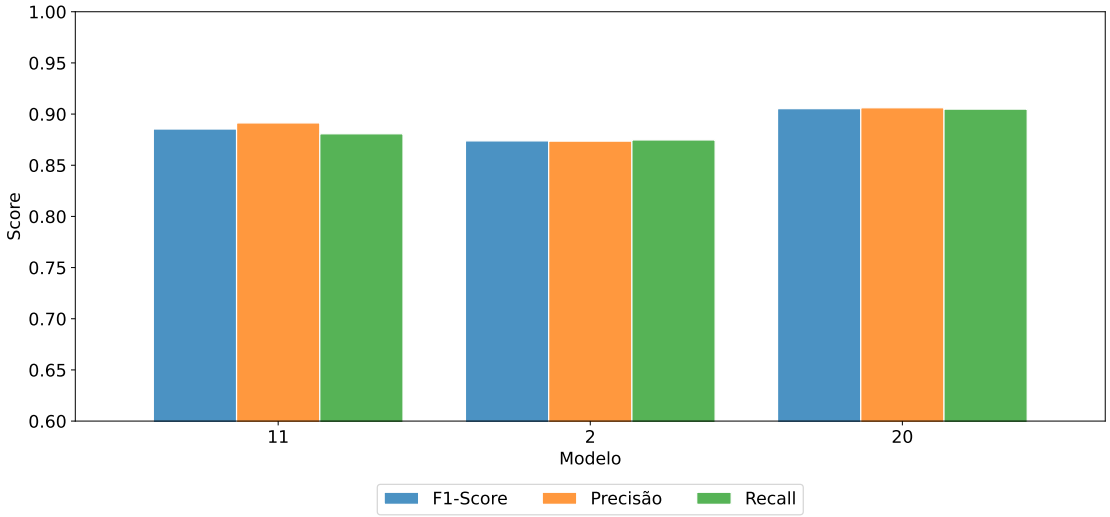
Figura 6.4 – F1-Score das entidades dos modelos 1, 10 e 19



Fonte: O Autor

Os modelos do grupo de amostras 2 apresentam um comportamento semelhante aos modelos do grupo de amostras 1, conforme mostrado na Figura 6.5. O modelo 20 possui melhor *recall* e precisão do que os modelos do mesmo grupo, resultando em um F1-score superior, sugerindo o efeito benéfico de um *dropout* maior.

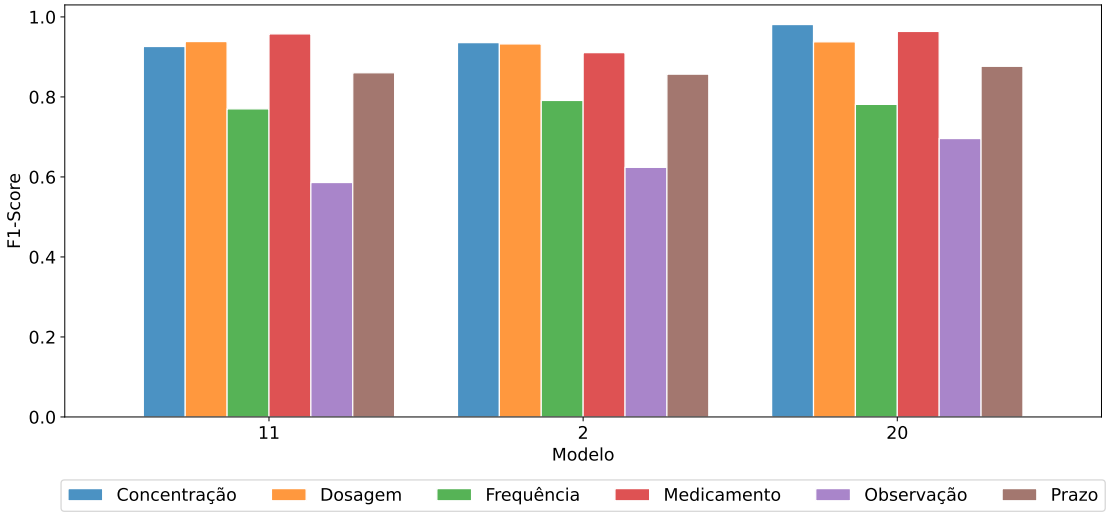
Figura 6.5 – Métricas dos modelos 2, 11 e 20. A escala do eixo *Score* se inicia em 0.6



Fonte: O Autor

Mais uma vez, a análise das entidades confirma a indicação encontrada na avaliação média dos modelos. Nesse caso, todas as entidades do modelo 20 apresentam um F1-score superior em comparação com as entidades equivalentes dos outros dois modelos.

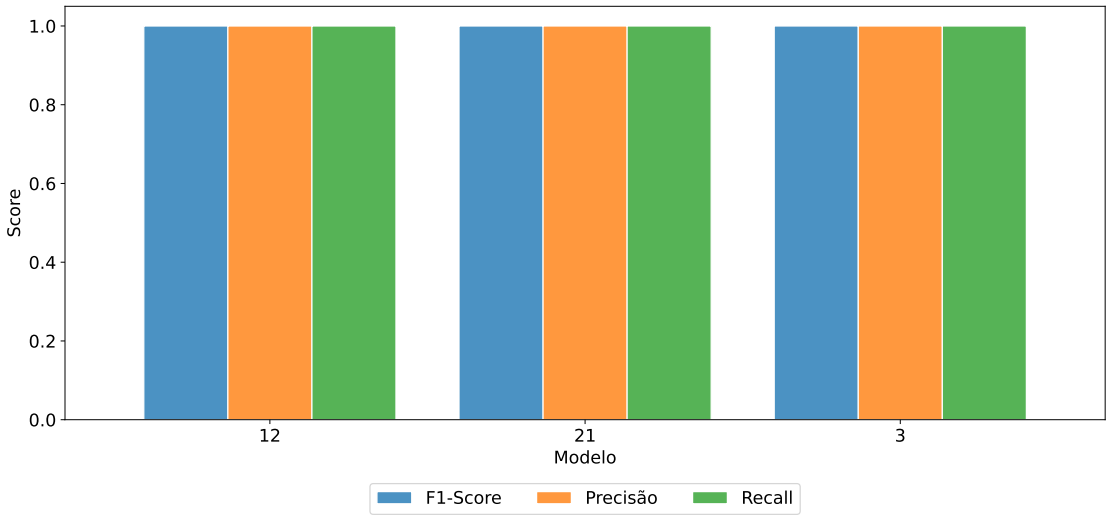
Figura 6.6 – F1-Score das entidades dos modelos 2, 11 e 20



Fonte: O Autor

A Figura 6.7 mostra as métricas dos modelos pertencentes ao grupo de amostras 3, onde todas as métricas estão muito próximas de 1, o que indica um desempenho quase perfeito para esses modelos. Isso era esperado, dado que o grupo de amostras 3 foi projetado para treinar os modelos para reconhecer diversos formatos de texto para cada entidade, incluindo textos com erros de digitação. Neste caso, alterar os valores *dropout* não tem um efeito visível.

Figura 6.7 – Métricas dos modelos 3, 12 e 21



Fonte: O Autor

Em resumo, a análise de desempenho versus *dropout* revelou que uma taxa maior de *dropout* pode melhorar a capacidade de generalização de modelos. Isso não se aplicou aos modelos do grupo de amostra 3, mas foi observado nos modelos dos grupos de amostras 1 e 2, onde os modelos de maior taxa de *dropout* exibiram o melhor desempenho em

termos de precisão, *recall* e *F1-score*. Esses resultados sugerem que, ao ajustar o valor do *dropout*, é possível otimizar o desempenho dos modelos com poucas amostras e obter resultados mais robustos e generalizáveis.

6.1.3 Desempenho em relação ao *batch size*

Nesta análise, o objetivo é comparar o desempenho dos modelos considerando diferentes valores de *batch size*. A comparação será realizada com os modelos que apresentam uma configuração de *dropout* igual a 0. Semelhante à avaliação anterior, os modelos serão agrupados por grupos de amostras.

Tabela 6.3 – Modelos selecionados para a avaliação do desempenho em relação ao *batch size*

Modelo	*Dropout*	*Batch size*	Grupo
1	0.0	4	1
2	0.0	4	2
3	0.0	4	3
4	0.0	32	1
5	0.0	32	2
6	0.0	32	3
7	0.0	64	1
8	0.0	64	2
9	0.0	64	3

Fonte: O Autor

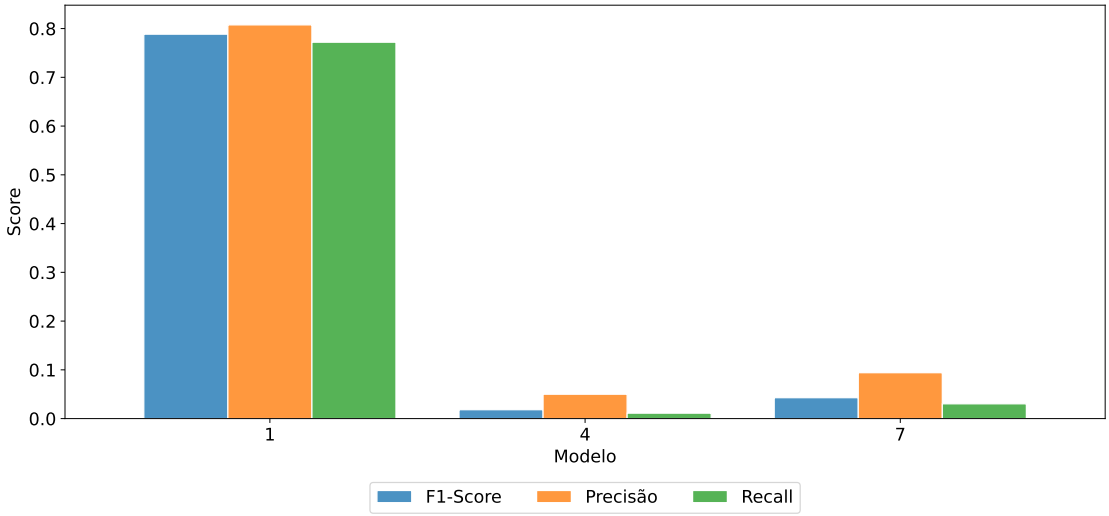
- Grupo de amostras 1: foram selecionados os modelos 1, 4 e 7.
- Grupo de amostras 2: foram selecionados os modelos 2, 5 e 8.
- Grupo de amostras 3: foram selecionados os modelos 3, 6 e 9.

Diferentemente das análises anteriores, nesta avaliação é possível extrair conclusões diretamente das médias gerais dos modelos. Aumentar o *batch size* pode prejudicar o desempenho com conjuntos de dados menores. Esse efeito ocorre quando o *batch size* é desproporcionalmente grande em relação ao tamanho do conjunto de dados, levando a uma generalização inadequada do modelo.

Dessa forma, para o grupo de amostras 1, não é recomendado utilizar um *batch size* igual ou superior a 32. Já para o grupo de amostras 2, não é aconselhável utilizar um *batch size* igual ou superior a 64.

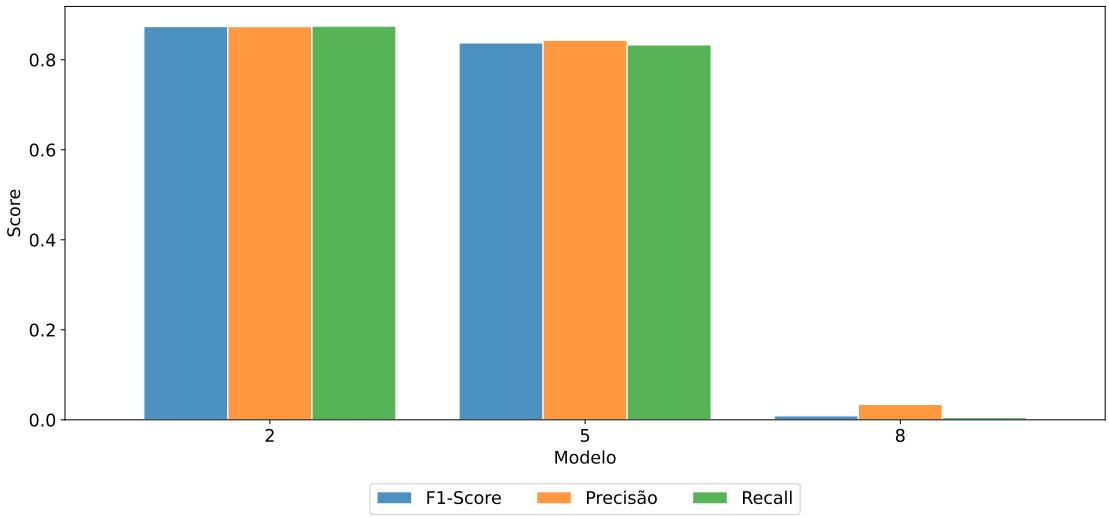
As Figuras 6.8, 6.9 e 6.10 ilustram essa constatação:

Figura 6.8 – Métricas dos modelos 1, 4 e 7



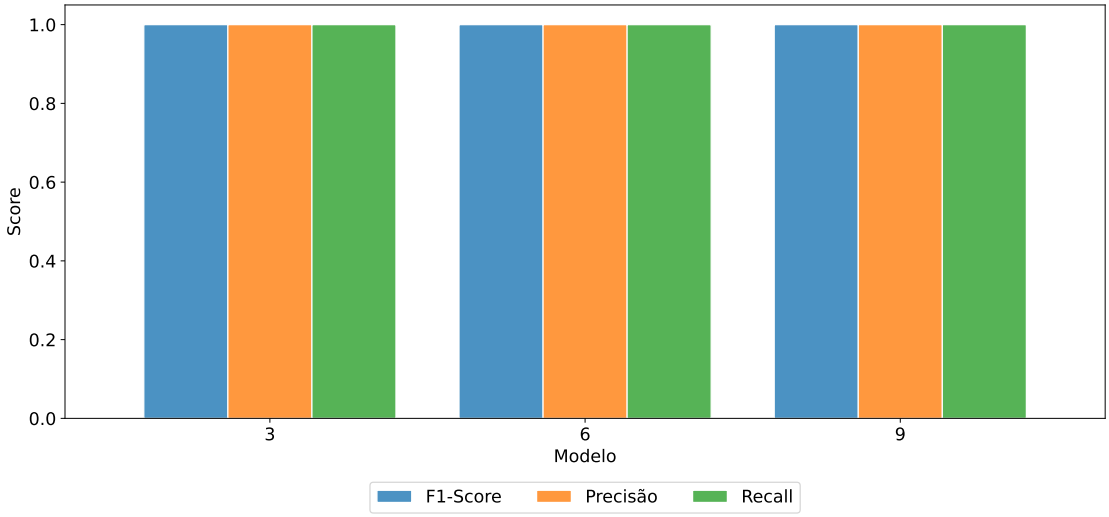
Fonte: O Autor

Figura 6.9 – Métricas dos modelos 2, 5 e 8



Fonte: O Autor

Figura 6.10 – Métricas dos modelos 3, 6 e 9



Fonte: O Autor

6.1.4 Seleção dos modelos para treinamento final

Levando em conta as avaliações realizadas, optou-se por selecionar todos os modelos do grupo de amostras 3, além do modelo 20, pertencente ao grupo de amostras 2. Os modelos do grupo de amostras 1 foram descartados, uma vez que sua performance foi comprovadamente inferior em comparação aos modelos escolhidos. A Tabela 6.4 apresenta detalhadamente os modelos selecionados:

Tabela 6.4 – Modelos selecionados para treinamento final

Modelo	*Dropout*	*Batch size*	Grupo
3	0.0	4	3
6	0.0	32	3
9	0.0	64	3
12	0.2	4	3
15	0.2	32	3
18	0.2	64	3
20	0.5	4	2
21	0.5	4	3
24	0.5	32	3
27	0.5	64	3

Fonte: O Autor

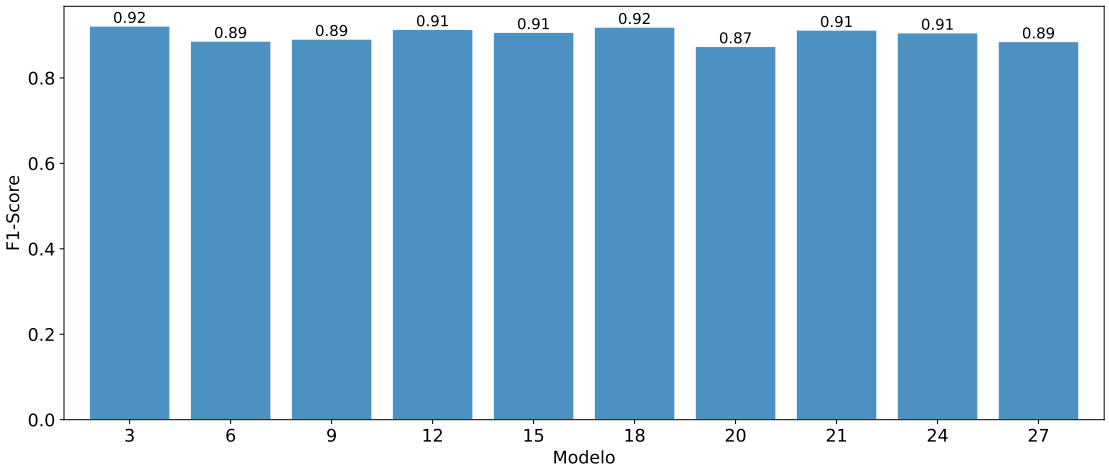
6.2 Avaliação dos modelos selecionados em conjunto de teste

Para obter o modelo final, os modelos selecionados na validação cruzada passaram por uma nova rodada de treinamento, utilizando todas as amostras disponíveis em seus respectivos grupos. Para avaliar o desempenho final, os modelos foram submetidos a um mesmo conjunto de teste.

O conjunto de teste foi gerado artificialmente com textos semelhantes ao do grupo de amostras 3, utilizado no treinamento. Assim, conseguimos avaliar os modelos do grupo 3 para o que exatamente eles foram propostos. Além disso, pode-se avaliar a capacidade do modelo 20, pertencente ao grupo 2, de generalizar nomes de medicamentos compostos e concentrações compostas, por exemplo, ao mesmo tempo que este modelo não foi exposto a entidades com este padrão.

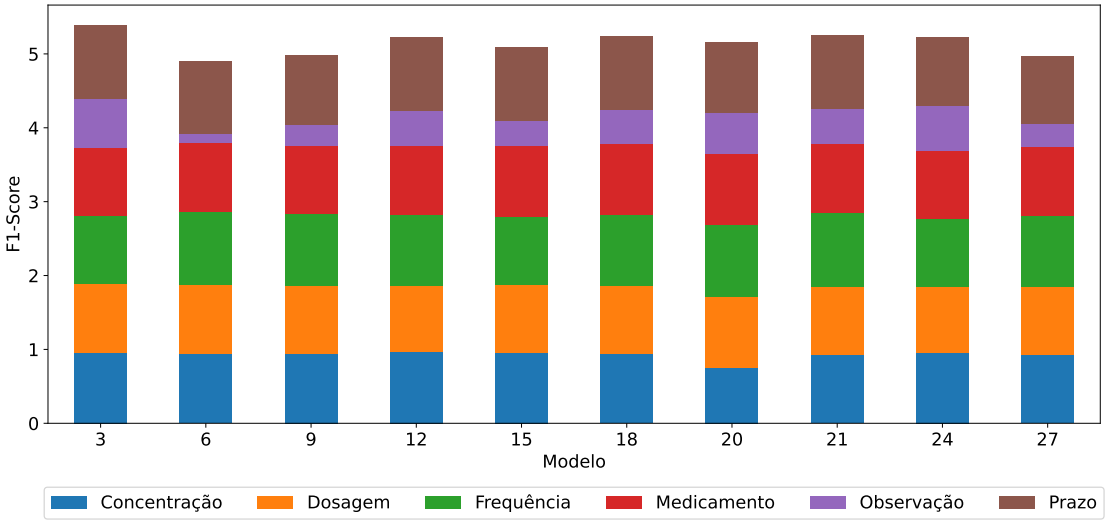
A Figura 6.11 exibe o desempenho dos modelos no conjunto de teste. Surpreendentemente, todos os modelos do grupo de amostras 3 apresentaram desempenho semelhante, exceto pelo modelo 3, que se destacou entre todos. A análise por entidade pode ser vista na Figura 6.12, onde fica evidente um melhor desempenho do modelo 3, destacando-se especialmente a entidade **OBSERVACAO**, que possui um desempenho significativamente inferior nos demais modelos.

Figura 6.11 – F1-Score dos modelos selecionados



Fonte: O Autor

Figura 6.12 – F1-Score das entidade dos modelos selecionados



Fonte: O Autor

Entretanto, ao analisar a matriz de confusão normalizada do modelo 3, na Figura 6.13, o modelo apresenta um desempenho aquém do esperado para a entidade **MEDICAMENTO**.

Diante disso, optou-se por selecionar outro modelo como o modelo final. As principais entidades em uma prescrição médica são **MEDICAMENTO**, **CONCENTRACAO** e **DOSAGEM**. Portanto, decidiu-se priorizar os modelos com melhor desempenho nessas entidades, principalmente em **MEDICAMENTO**, mesmo que isso implique em um desempenho inferior em entidades como **OBSERVACAO**. Entende-se que é possível lidar com reconhecimentos errôneos dessas outras entidades por meio de pós-processamento. Assim, o modelo final escolhido foi o modelo 18, com a matriz de confusão representada na Figura 6.14.

Figura 6.13 – Matriz de confusão do modelo 3

Fonte: O Autor

Figura 6.14 – Matriz de confusão do modelo 18

Fonte: O Autor

6.3 Testes da API

O modelo 18, selecionado como modelo final, pertence ao grupo de amostras 3. Para testar a API, as prescrições médicas coletadas foram submetidas, já que apenas auxiliaram na criação do grupo de amostras 3, mas não fazem parte dele. Entende-se que a estrutura das prescrições médicas coletadas possa enviesar o resultado, mas acredita-se ser válido, uma vez que a maioria das prescrições médicas apresenta estrutura semelhante. Segue um exemplo da utilização da API, onde a Figura 6.15 mostra o texto extraído do arquivo de entrada PDF e a Figura 6.16 mostra o resultado final retornado pela API:

Figura 6.15 – Exemplo 1 - Texto extraído

```
Uso oral: {newline} 1  Sinotr 400mg/5 ml {newline} Dar 5,5 ml de 12/12 h, 10 dias
{newline} Uso Interno: {newline}   Azitromicina 200mg / 5ml
```

Fonte: O Autor

Figura 6.16 – Exemplo 1 - Informações retornadas pela API

Medicamento	Concentração	Dosagem	Frequência	Prazo	Observação
Sinotr	400mg/5ml	5,5ml	24 horas	10 dias	Nenhuma

Fonte: O Autor

Como é possível observar, algumas padronizações foram feitas. Caso alguma das entidades abaixo não seja identificada para determinado medicamento, assume-se valores padrões:

- **CONCENTRACAO** = Não informada
- **DOSAGEM** = Não informada
- **FREQUENCIA** = 24 horas
- **PRAZO** = Indeterminado
- **OBSERVACAO** = Nenhuma

Abaixo, segue outro exemplo mais complexo, onde o valor encontrado para a **FREQUENCIA** é convertido em horas. Ex: 2xx se torna 12 horas.

Figura 6.17 – Exemplo 2 - Texto extraído

```
Uso Interno: {newline}  Azitromicina 200mg / 5ml.............  1 vidro grande ou
2 pequenos {newline} Administrar 5 ml (cinco) , 1x dia, por 5 dias. {newline}
Prednisolona 3mg/ml xarope..........................  1 vidro {newline}Administrar
6ml via oral 1x dia por 5 dias. {newline} Hoje em qualquer hora e a partir de
amanha pela manha. {newline}  Cloridrato de Bromexina xarope pediatrico.....
1 vidro {newline} Administrar 4ml via oral 2xx dia por 7 dias. {newline}
```

Fonte: O Autor

Figura 6.18 – Exemplo 2 - Informações retornadas pela API

Medicamento	Concentração	Dosagem	Frequência	Prazo	Observação
Azitromicina	200mg/5ml	5ml	24 horas	5 dias	Nenhuma
Prednisolona	3mg/ml	6ml	24 horas	5 dias	Hoje em qualquer hora e a partir de amanha pela manha
Cloridrato de Bromexina	Não informada	4ml	12 horas	7 dias	Nenhuma

Fonte: O Autor

Desta forma, os resultados dos testes indicam que a ferramenta de extração de dados em prescrições médicas é viável para uso prático até o momento. No entanto, são necessários testes adicionais com amostras de outros Conselhos Regionais de Medicina ou operadoras de saúde, por exemplo, para validar a eficácia da API em diferentes contextos e garantir a precisão e confiabilidade dos resultados. Ao expandir os testes e ajustar o modelo com base em novos dados e *feedback*, a ferramenta tem potencial para melhorar ainda mais.

7 DISCUSSÃO

Neste capítulo, serão discutidos os resultados obtidos ao longo deste trabalho, analisando criticamente o desempenho do modelo selecionado e a eficácia da API desenvolvida. Além disso, serão abordadas as limitações encontradas na pesquisa e sugeridas possibilidades de melhoria para aprimorar ainda mais a solução proposta. O objetivo desta discussão é fornecer uma visão completa do que foi alcançado e identificar áreas onde futuras pesquisas e desenvolvimentos podem ser aplicados.

7.1 Análise crítica dos resultados

Os resultados obtidos ao longo deste trabalho demonstram a eficácia da ferramenta para a extração de informações relevantes de prescrições médicas. O modelo 18, com *dropout* de 0.2 e *batch size* de 64, pertencente ao terceiro grupos de amostras, apresentou desempenho satisfatório nas entidades mais importantes, como **MEDICAMENTO**, **CONCENTRACAO** e **DOSAGEM**. A escolha de priorizar essas entidades foi baseada na premissa de que são os elementos mais importantes em uma prescrição médica.

No entanto, a entidade **OBSERVACAO** foi a única entre todas que apresentou um desempenho muito abaixo do esperado. Entende-se que o desempenho está relacionado ao fato desta entidade ser, na maioria do casos, uma frase ou sentença com muitas variações em sua estrutura. Assim, para identificar corretamente as observações presentes nas prescrições médicas, seria necessário usar uma abordagem diferente do NER.

Os testes realizados com a API desenvolvida mostraram resultados promissores, que indicam estar pronta para ser utilizada em um aplicativo, conforme a finalidade inicial do trabalho.

7.2 Limitações e possibilidades de melhoria

Embora os resultados apresentados sejam promissores, existem algumas limitações que devem ser abordadas.

- **Diversidade de dados**: Os dados utilizados neste trabalho provêm basicamente de uma única fonte, o CREMERS, e possuem estrutura semelhante. Para melhorar a generalização do modelo, seria benéfico incluir dados de diferentes fontes, como

outros Conselhos Regionais de Medicina e operadoras de saúde.

- **Atualização contínua do modelo**: O modelo deve ser continuamente atualizado com novos dados e *feedback* dos usuários, permitindo aprimorar sua capacidade de lidar com diferentes cenários e evoluir conforme as necessidades do setor de saúde.
- **Adaptação para outras línguas**: O modelo atual é treinado especificamente para o idioma português. No entanto, uma possibilidade de melhoria seria adaptar o modelo para trabalhar com prescrições médicas em outros idiomas, aumentando sua aplicabilidade.
- **Expansão para comandos de voz**: Outra possibilidade de melhoria seria a adaptação do modelo para lidar com comandos de voz. Isso permitiria que os usuários ditassem as prescrições médicas, que seriam automaticamente convertidas em texto e extraídas pela API. Essa funcionalidade poderia aumentar a eficiência e a praticidade do processo de prescrição.
- **Processamento de imagens**: A integração de técnicas de processamento de imagens e reconhecimento óptico de caracteres (OCR) no modelo também apresenta potencial para expandir seu escopo de aplicação. Isso possibilitaria a extração de informações de prescrições médicas a partir de imagens digitalizadas ou fotografias, ampliando as fontes de dados utilizáveis e facilitando a digitalização de prescrições médicas em papel.

Ambas as expansões propostas, comandos de voz e processamento de imagens, têm o potencial de aumentar significativamente a utilidade e a abrangência do modelo. Ao incorporar essas funcionalidades, o sistema proposto poderia se tornar uma solução mais completa e versátil para a análise de informações de prescrições médicas, contribuindo ainda mais para a qualidade e a eficiência do processo.

Em suma, este trabalho apresenta resultados encorajadores na extração de informações de prescrições médicas, mas há espaço para aprimoramento e expansão da solução proposta. Abordar as limitações mencionadas e explorar possibilidades de melhoria pode resultar em uma ferramenta ainda mais valiosa para o setor de saúde.

8 CONCLUSÃO

Ao longo deste trabalho, foi desenvolvida uma ferramenta de extração de dados em prescrições médicas utilizando aprendizado de máquina e processamento de linguagem natural. O processo envolveu a coleta e pré-processamento das amostras, rotulagem das amostras, geração de novas amostras, treinamento do modelo NER do SpaCy e desenvolvimento da API. A ferramenta desenvolvida se mostrou eficaz na extração das entidades presentes nas prescrições médicas e contribuiu para tornar o processo de extração mais prático e ágil para os usuários.

Com base nos resultados alcançados, é possível afirmar que existem perspectivas de continuidade para este trabalho. Uma possibilidade seria a expansão da ferramenta para que ela possa ser utilizada com comando de voz e processamento de imagens, se tornando uma solução mais completa.

Em síntese, a ferramenta desenvolvida neste trabalho se mostrou eficaz na extração de informações relevantes em prescrições médicas, demonstrando a aplicabilidade do uso de técnicas de processamento de linguagem natural e aprendizado de máquina neste contexto.

REFERÊNCIAS

Amazon Web Services, Inc. **Amazon Comprehend Medical**. 2022. <https://aws.amazon.com/comprehend/medical/>. Acessado em 29 de março de 2023.

ANVISA. **DADOS_ABERTOS_MEDICAMENTOS.csv**. 2023. <https://dados.anvisa.gov.br/dados/DADOS_ABERTOS_MEDICAMENTOS.csv>. Acesso em: 29 mar. 2023.

ANVISA. **Lista A de medicamentos incluídos (10/02/2023)**. 2023. <https://www.gov.br/anvisa/pt-br/setorregulado/regularizacao/medicamentos/medicamentos-de-referencia/arquivos/lista-a-incluidos-10022023.pdf/view>. Acesso em: 29 mar. 2023.

ANVISA. **Lista B de medicamentos incluídos (10/02/2023)**. 2023. <https://www.gov.br/anvisa/pt-br/setorregulado/regularizacao/medicamentos/medicamentos-de-referencia/arquivos/lista-b-incluidos-10022023.pdf/view>. Acesso em: 29 mar. 2023.

AQUARELA. **Aprendizado de máquina: subáreas e aplicações**. 2022. <https://www.aquare.la/aprendizado-de-maquina-subareas-e-aplicacoes/>. Acesso em: 29 mar. 2023.

BISHOP, C. M. **Pattern Recognition and Machine Learning**. New York, USA: Springer, 2006.

BOTTOU, L. Large-scale machine learning with stochastic gradient descent. In: **Proceedings of COMPSTAT**. [S.l.: s.n.], 2010. p. 177–186.

CHUI, M. et al. **The state of AI in 2022 - and a half decade in review**. 2022. <https://www.mckinsey.com/business-functions/mckinsey-analytics/our-insights/the-state-of-ai-in-2022-and-a-half-decade-in-review>. Acesso em: 29 mar. 2023.

GOODFELLOW, I.; BENGIO, Y.; COURVILLE, A. **Deep Learning**. Cambridge, Massachusetts, USA: MIT Press, 2016.

HONNIBAL, M.; MONTANI, I. **spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing**. 2017. <https://spacy.io>. Acesso em 29 mar. de 2023.

HUTTER, F.; KOTTHOFF, L.; VANSCHOREN, J. **Automated Machine Learning: Methods, Systems, Challenges**. Cham, Switzerland: Springer, 2019.

IBM Corporation. **IBM Global AI Adoption Index 2022**. 2022. <https://www.ibm.com/downloads/cas/GVAGA3JP>. Acesso em: 29 mar. 2023.

JURAFSKY, D.; MARTIN, J. H. **Speech and Language Processing**. 3. ed. Nova Jersey, USA: Prentice Hall, 2019.

KARIMI, S.; METKE-JIMENEZ, A.; KEMP, M. Cadec: A corpus of adverse drug event annotations. **Journal of Biomedical Informatics**, v. 55, p. 73–81, 2015.

LAMPLE, G. et al. Neural architectures for named entity recognition. In: **Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies**. [S.l.: s.n.], 2016. p. 260–270.

MITCHELL, T. M. **Machine Learning**. New York, USA: McGraw-Hill, 1997.

MOENS, M.-F. **Information Extraction: Algorithms and Prospects in a Retrieval Context**. London, United Kingdom: Springer, 2006.

NADEAU, D.; SEKINE, S. A survey of named entity recognition and classification. **Lingvisticae Investigationes**, v. 30, n. 1, p. 3–26, 2007.

NEVES, M.; TAGGER, B. **MedaCy: Medical Text Mining and Information Extraction with spaCy**. 2022. <https://github.com/NanoNLP/medaCy>. Acessado em 29 de março de 2023.

PRECHELT, L. Early stopping - but when? In: ORR, G.; MüLLER, K.-R. (Ed.). **Neural Networks: Tricks of the Trade**. Berlin, Heidelberg, Germany: Springer, 1998. p. 55–69.

Presidência da República. **Lei nº 13.715, de 24 de setembro de 2018**. 2018. <http://www.planalto.gov.br/ccivil_03/_Ato2015-2018/2018/Lei/L13715.htm>. Acesso em: 29 mar. 2023.

SANG, E. F. T. K.; MEULDER, F. D. Introduction to the conll-2003 shared task: Language-independent named entity recognition. In: **Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003**. [S.l.: s.n.], 2003. v. 4, p. 142–147.

SOKOLOVA, M.; LAPALME, G. A systematic analysis of performance measures for classification tasks. **Information Processing & Management**, v. 45, n. 4, p. 427–437, 2009.

SOYSAL, E. et al. Clamp – a toolkit for efficiently building customized clinical natural language processing pipelines. **Journal of the American Medical Informatics Association**, v. 25, n. 3, p. 331–336, 2018.

SPACY. **Processing pipelines**. 2023. <https://spacy.io/usage/processing-pipelines>. Acesso em: 29 mar. 2023.

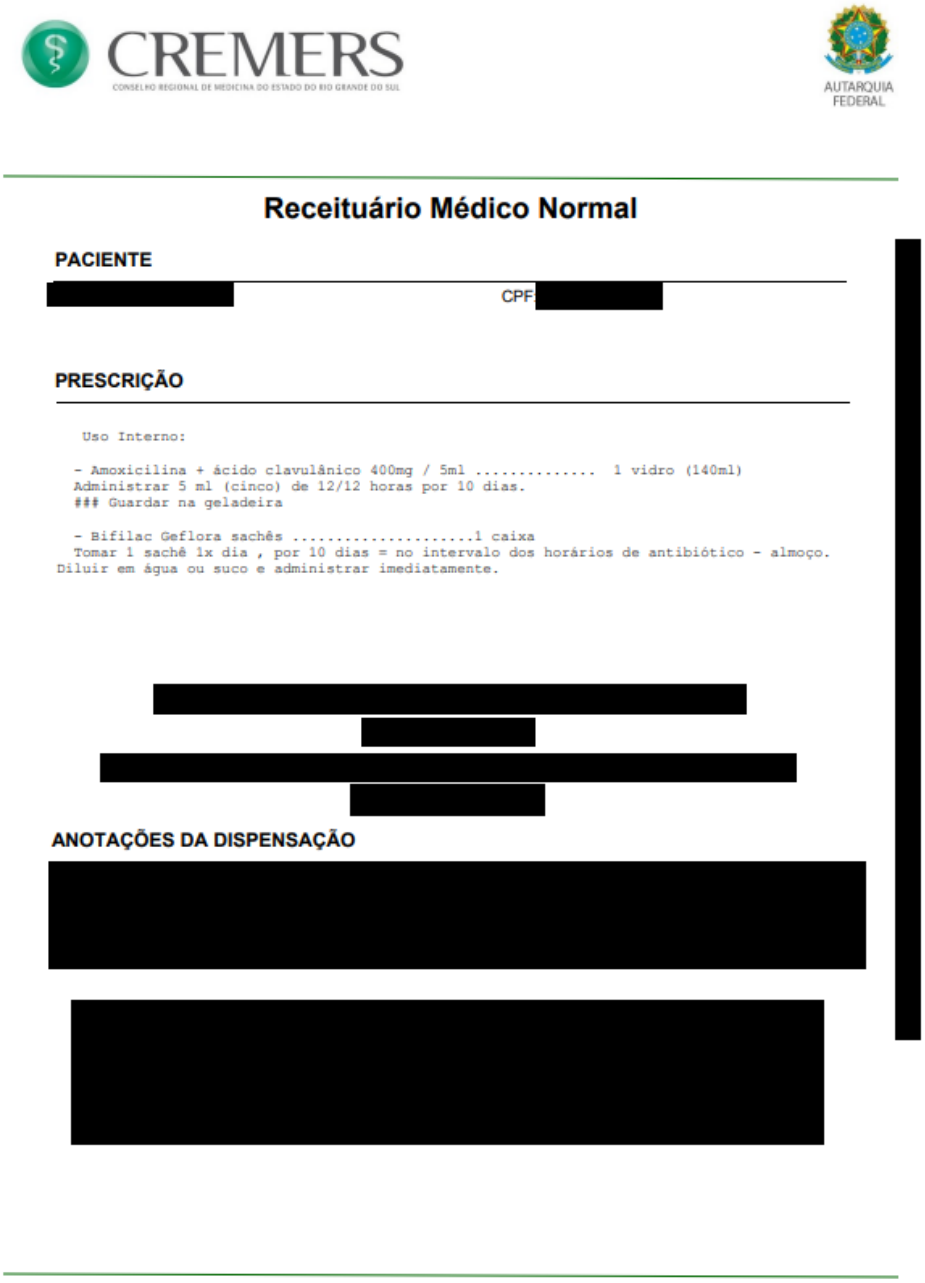
SRIVASTAVA, N. et al. Dropout: A simple way to prevent neural networks from overfitting. **Journal of Machine Learning Research**, v. 15, p. 1929–1958, 2014.

YOSINSKI, J. et al. How transferable are features in deep neural networks? In: **Advances in Neural Information Processing Systems**. [S.l.: s.n.], 2014. v. 27, p. 3320–3328.

ZELLE. **Zelle: cuidado dos filhos**. 2023. <https://www.zelle.com.br>. Acesso em: 29 mar. 2023.

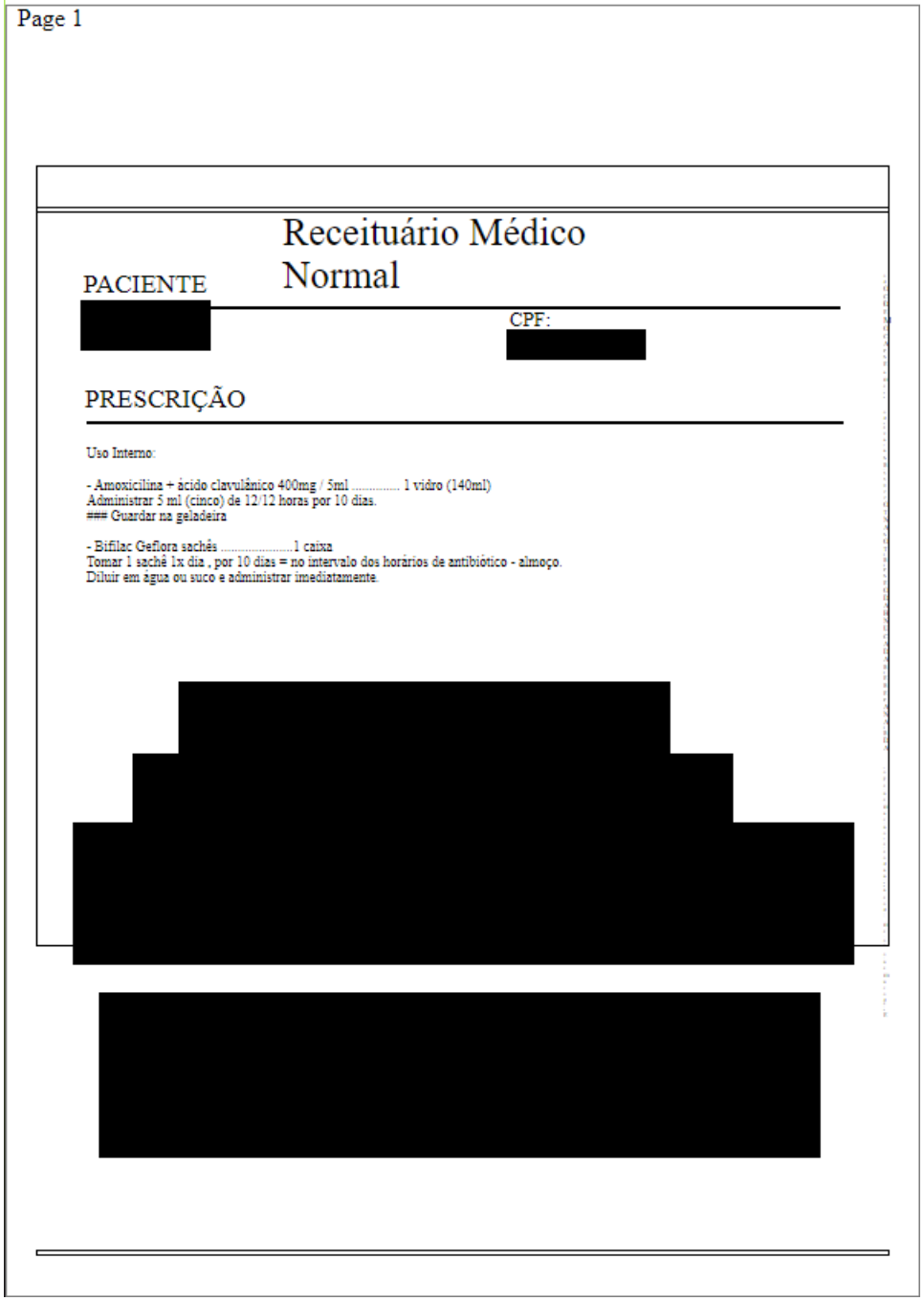
APÊNDICE A — PRÉ-PROCESSAMENTO

Figura A.1 – Prescrição médica eletrônica emitida em PDF



Fonte: O Autor

Figura A.2 – HTML convertido da Prescrição médica eletrônica emitida em PDF



Fonte: O Autor

APÊNDICE B — CONFIGURAÇÕES DOS MODELOS

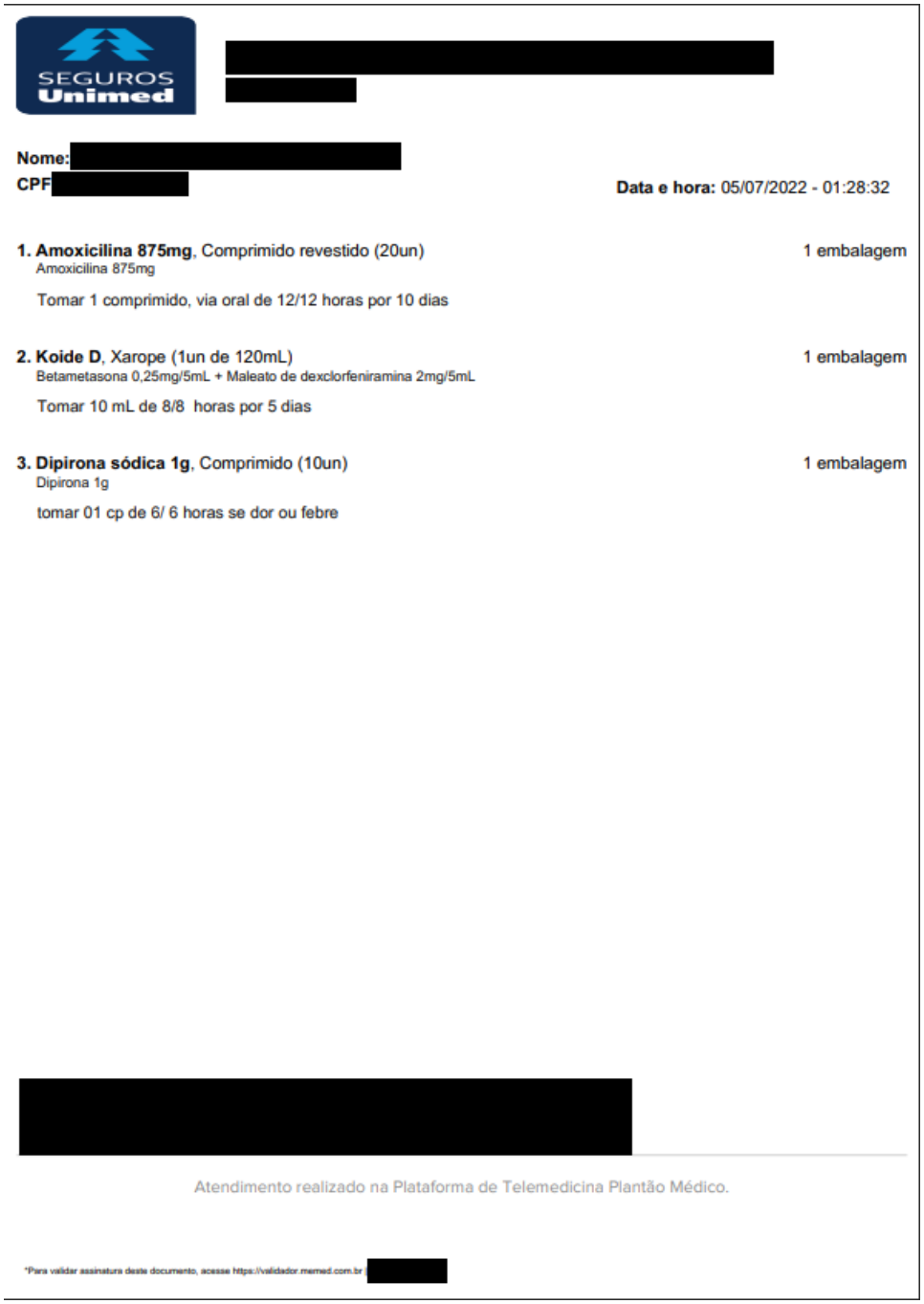
Tabela B.1 – Configurações dos modelos - os grupos se referem ao grupos de amostras, conforme descrito na subseção 5.3

Modelo	*Dropout*	*Batch size*	Grupo
1	0.0	4	1
2	0.0	4	2
3	0.0	4	3
4	0.0	32	1
5	0.0	32	2
6	0.0	32	3
7	0.0	64	1
8	0.0	64	2
9	0.0	64	3
10	0.2	4	1
11	0.2	4	2
12	0.2	4	3
13	0.2	32	1
14	0.2	32	2
15	0.2	32	3
16	0.2	64	1
17	0.2	64	2
18	0.2	64	3
19	0.5	4	1
20	0.5	4	2
21	0.5	4	3
22	0.5	32	1
23	0.5	32	2
24	0.5	32	3
25	0.5	64	1
26	0.5	64	2
27	0.5	64	3

Fonte: O Autor

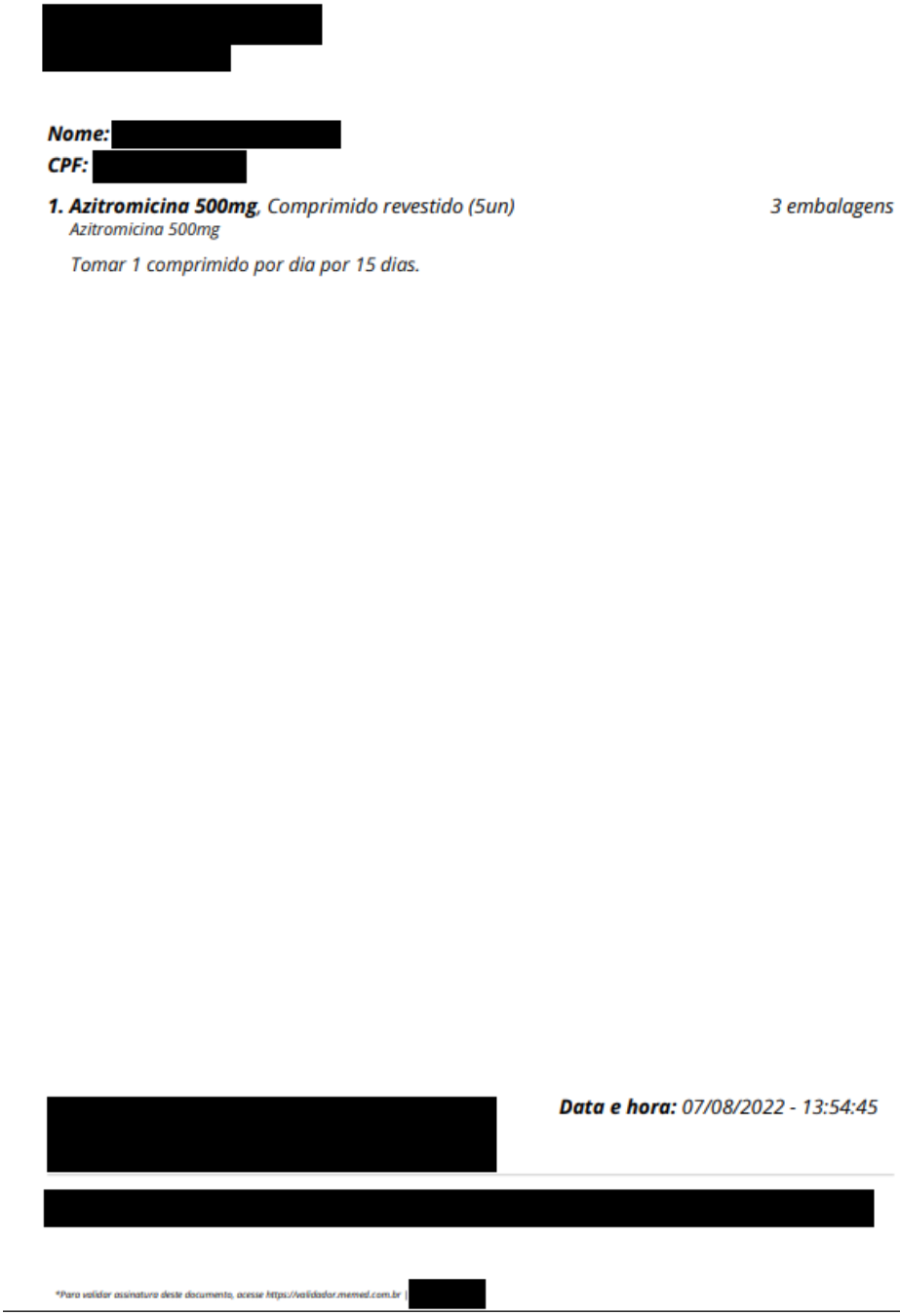
ANEXO A — EXEMPLOS DE AMOSTRAS COLETADAS

Figura A.1 – Exemplo 1 - Prescrição médica emitida pelo Unimed



Fonte: O Autor

Figura A.2 – Exemplo 2 - Prescrição médica emitida pela plataforma MEMED


[redacted]

Nome: [redacted]
CPF: [redacted]

1. Azitromicina 500mg, *Comprimido revestido (5un)* — *3 embalagens*
Azitromicina 500mg

Tomar 1 comprimido por dia por 15 dias.

[redacted]

Data e hora: *07/08/2022 - 13:54:45*

[redacted]

**Para validar assinatura deste documento, acesse https://validador.memed.com.br |* [redacted]

Fonte: O Autor