

## RESUMO

Este documento tem por finalidade apresentar uma abordagem para a estruturação automática de laudos médicos em português por meio da extração de informações clínicas utilizando processamento de linguagem natural (NLP) e aprendizado de máquina. Foram propostas duas arquiteturas, Bi-LSTM-CRF e CRF baseada em *features*, além da técnica de *bootstrapping*, utilizada para criação de conjunto de dados anotados a partir de um conjunto pequeno. As métricas *precision*, *recall* e *F1-score*, obtidas a partir do conjunto de teste, foram todas acima de 80%. As análises de robustez mostraram que a solução é promissora e poderá ser aprimorada. A explicabilidade dos modelos de CRF, que apresenta características que o algoritmo observou durante o treinamento, também auxiliou no entendimento do que foi realizado. As implementações são escaláveis e adaptáveis, não somente para novas modalidades médicas, mas também para outras tarefas dentro da empresa, o que não acontece com o microserviço de estruturação atual, baseado em expressões regulares. Além disso, os resultados mostraram uma potencial otimização de tempo de desenvolvimento e custo em produção, simplificando a estrutura em nuvem vigente.

**Palavras-chave:** Aprendizado de máquina. Laudos médicos. Extração de informações clínicas. Processamento de linguagem natural.

fossem treinados vários modelos durante um processo em paralelo. Para determinadas arquiteturas, existe ainda o *variational dropout* (GIORGI; BADER, 2019). A diferença é que a técnica se aplica a partes específicas da rede durante múltiplos passos.

Com relação à arquitetura CRF, têm-se outros algoritmos de otimização, apresentados no quadro 1. Nesse caso, podem utilizar a regularização L1 e/ou L2 ou algum método mais específico. Para o *Passive Aggressive*, por exemplo, existe um outro tipo de coeficiente que realiza esse papel.

Quadro 1 – Resumo dos principais algoritmos para a CRF.

| Algoritmo   | Sigla  | Características   |
|---|--------|---|
| <i>Limited-memory Broyden-Fletcher-Goldfarb-Shanno</i>    | L-BFGS | Maximiza o logaritmo da probabilidade dos dados de treino com regularização L1 e/ou L2. Melhora os pesos muito lentamente no início, mas converge rapidamente no final. |
| <i>Stochastic Gradient Descent with L2 regularization</i> | L2-SGD | Maximiza o logaritmo da probabilidade dos dados de treino com regularização L2. Aproxima dos pesos ótimos rapidamente, mas mostra convergência lenta no final.          |
| <i>Averaged Perceptron</i>                                | AP     | Calcula a média dos pesos em todas as atualizações do processo. É o algoritmo mais veloz no treinamento.  |
| <i>Passive Aggressive</i>                                 | PA     | Utiliza o <i>loss</i> para atualizar o modelo.  |
| <i>Adaptive Regularization Of Weight Vector</i>           | AROW   | Também utiliza o <i>loss</i> , porém o computa com uma formulação diferente.  |

Fonte – Elaborado pela autora com base em (OKAZAKI, 2007).

No presente projeto os diferentes otimizadores tiveram seu desempenho avaliado através de experimentos. Alguns dos parâmetros disponíveis foram variados para que seus efeitos no aprendizado do modelo fossem melhor compreendidos.

### 3.2 MÉTRICAS DE DESEMPENHO

Ao treinar um modelo de inteligência artificial é de suma importância avaliar o seu desempenho através de métricas. As mais clássicas, como acurácia, *precision*, *recall* e *F1-score*, são calculadas através da matriz de confusão (indicada na equação 6), que dispõe sobre o número de classificações de um modelo para cada categoria ou entidade e na qual TP são verdadeiros positivos, FP são falsos positivos, FN os falsos negativos e TN, verdadeiros negativos.

$$\begin{bmatrix} TP & FP \\ FN & TN \end{bmatrix}. \quad (6)$$

Para trabalhos que envolvem NLP, deve-se levar em conta ainda a forma de interpretação sobre o que é um acerto e um erro. Para reconhecimento de entidades, por exemplo, pode-se considerar uma predição correta como a extração da entidade exata,

Para melhor esclarecer a diferença entre os cálculos do Quadro 2, apresenta-se um exemplo. Supondo-se que é desejado identificar duas entidades clínicas em um conjunto de laudos médicos, representadas pelos códigos SNOMED-CT "A", "B", "C", é retornada a seguinte matriz de confusão:

Tabela 1 – Matriz de confusão para exemplo fictício de reconhecimento de entidades.

|          |   | Resposta Verdadeira |   |   |
|----------|---|---------------------|---|---|
|          |   | A                   | B | C |
| Predição | A | 4                   | 6 | 3 |
|          | B | 1                   | 2 | 0 |
|          | C | 1                   | 2 | 6 |

Fonte – Elaborado pela autora com base em (SHMUJELI, 2019).

Para as métricas micro, como observam-se todas as entidades ao mesmo tempo, cada predição errada é um falso positivo ou negativo, dependendo do ponto de vista. Por exemplo, se uma amostra para a entidade "A" é classificada como "B", então tal amostra pode ser vista como um falso positivo para "B" e falso negativo para "A". Já no caso da macro, o cálculo é feito para cada entidade da mesma maneira que em um caso binário, seguindo a matriz de confusão e posteriormente se obtém a média. Dessa forma, para o problema fictício, os resultados são apresentados na Tabela 2.

Tabela 2 – Métricas para exemplo fictício de reconhecimento de entidades.

|                  | Micro                                    | Macro                                  |
|------------------|--|--|
| <i>Recall</i>    | $\frac{4+2+6}{12+6+3+1+0+1+2} = 48\%$    | $\frac{0.667+0.20+0.667}{3} = 51.1\%$  |
| <i>Precision</i> | $\frac{4+2+6}{12+6+3+1+0+1+2} = 48\%$    | $\frac{0.308+0.667+0.667}{3} = 54.7\%$ |
| <i>F1-score</i>  | $2 * \frac{0.48*0.48}{0.48+0.48} = 48\%$ | $\frac{0.421+0.308+0.667}{3} = 46.5\%$ |

Fonte – Elaborado pela autora com base em (SHMUJELI, 2019).

Para o presente projeto, ambas as formas, micro e macro, serão contabilizadas. Como cada uma delas observa o modelo de IA sob um ponto de vista, é relevante compará-las, principalmente a métrica *F1-score*.