UNIVERSIDADE FEDERAL DA PARAÍBA
CENTRO DE INFORMÁTICA
CURSO DE CIÊNCIA DE DADOS E INTELIGÊNCIA ARTIFICIAL

Gabrielly Silva Batista

Jéssica Chaves Nagahama

Classificador de Sintomas

JOÃO PESSOA
2025

Gabrielly Silva Batista

Jéssica Chaves Nagahama

Classificador de Sintomas

Relatório Final de Pesquisa apresentado ao Curso de Bacharelado em Ciência de Dados e Inteligência Artificial da Universidade Federal da Paraíba como requisito parcial para a conclusão da disciplina de Processamento de Linguagem Natural, sob orientação do Prof Yuri de Almeida Malheiros Barbosa.

JOÃO PESSOA

2025

SUMÁRIO

1 APRESENTAÇÃO DO PROBLEMA

A crescente disponibilidade de informações médicas e de saúde on-line tem gerado um volume massivo de dados textuais, abrangendo desde registros de pacientes até discussões em fóruns e plataformas de perguntas e respostas. Embora essa riqueza de informações represente um potencial significativo para aprimorar o diagnóstico, o tratamento e a pesquisa médica, sua natureza não estruturada apresenta desafios consideráveis para a extração eficiente de conhecimento. Nesse contexto, a classificação automática de sintomas a partir de perguntas formuladas por pacientes ou usuários torna-se uma tarefa de grande relevância.

A identificação precisa da especialidade médica mais adequada para tratar um conjunto específico de sintomas é crucial para otimizar o fluxo de atendimento, reduzir o tempo de espera e garantir que os pacientes sejam encaminhados aos profissionais mais qualificados. Métodos tradicionais, baseados em busca por palavras-chave ou regras predefinidas, frequentemente falham em capturar a complexidade da linguagem natural, a variedade de formas de expressar os sintomas e as nuances contextuais que podem ser essenciais para uma classificação correta.

Diante desse cenário, o projeto final da disciplina propõe a implementação de um classificador de sintomas utilizando técnicas de Processamento de Linguagem Natural (PLN) e modelos de aprendizado profundo de máquina. A abordagem visa desenvolver um sistema capaz de analisar a coluna "question" do dataset MedSquad, que contém perguntas formuladas por usuários, e atribuir a especialidade médica correspondente, representada na coluna "focus_area". Espera-se que a solução contribua para automatizar e aprimorar o processo de triagem e encaminhamento de pacientes, facilitando o acesso a informações médicas relevantes e promovendo uma assistência à saúde mais eficiente.

2 OBJETIVOS

O presente trabalho tem como objetivo geral desenvolver e implementar um sistema de

classificação automática de sintomas médicos utilizando técnicas de Processamento de Linguagem Natural (PLN). Especificamente, busca-se construir um modelo computacional capaz de analisar perguntas formuladas por usuários em linguagem natural e associá-las à especialidade médica mais adequada para o tratamento ou investigação dos sintomas descritos.

Para alcançar o objetivo geral, os seguintes objetivos específicos foram definidos:

1. Pré-processar e analisar o conjunto de dados MedSquad: Esta etapa envolve a limpeza, organização e transformação dos dados textuais contidos na coluna "question", bem como a análise da distribuição das especialidades médicas na coluna "focus_area".
2. Implementar e treinar um modelo de aprendizado de máquina: Serão explorados e implementados algoritmos de classificação, incluindo modelos de PLN, utilizando técnicas de aprendizado profundo, visando otimizar o desempenho na tarefa de classificação de sintomas.
3. Avaliar o desempenho do modelo: Métricas de avaliação relevantes, como acurácia, precisão, recall e F1-score, foram utilizadas para quantificar a eficácia do modelo em classificar corretamente as perguntas nas respectivas especialidades médicas.
4. Analisar os resultados e identificar possíveis melhorias: Os resultados obtidos serão analisados, buscando identificar padrões de acerto e erro do modelo, bem como possíveis direções para aprimorar o sistema de classificação.

Espera-se que o sistema desenvolvido contribua para a automatização e aprimoramento do processo de triagem e encaminhamento de pacientes, facilitando o acesso a informações médicas relevantes e promovendo uma assistência à saúde mais eficiente e ágil.

3 DADOS UTILIZADOS E PRÉ PROCESSAMENTO DOS DADOS

O conjunto de dados utilizado foi carregado inicialmente em um DataFrame do Pandas, com o objetivo de facilitar a manipulação e o pré-processamento dos dados textuais e categóricos. Primeiramente, o dataframe foi filtrado para conter os campos necessários para

efetuação da pesquisa, posteriormente, para assegurar a integridade dos dados, as linhas contendo valores ausentes (NaN) foram removidas, gerando um novo DataFrame, 'df_pares_sem_nan'. A remoção desses valores foi documentada através da impressão dos shapes do DataFrame antes e depois da operação, permitindo verificar a quantidade de dados excluídos.

Uma análise da distribuição das classes na coluna 'especialidade' foi realizada, utilizando a função value_counts(), para identificar possíveis desequilíbrios. Para diminuir o impacto de classes minoritárias, especialidades com frequência inferior a 7 foram agrupadas na categoria 'Other specialty'. Essa decisão teve como objetivo aprimorar o desempenho do modelo, evitando que classes com poucos exemplos prejudicassem o treinamento.

A limpeza do texto na coluna 'texto_sintoma' foi realizada por meio da aplicação da função limpar_texto. Essa função converte o texto para minúsculas, removeu a pontuação e eliminou as stopwords do idioma inglês, utilizando a lista fornecida pela biblioteca NLTK. O resultado da limpeza foi armazenado em uma nova coluna, 'texto_sintoma_limpo', permitindo a comparação entre o texto original e o texto limpo.

O conjunto de dados resultante foi então dividido em conjuntos de treinamento, validação e teste, utilizando a função train_test_split da biblioteca scikit-learn. Essa divisão estratégica, com uma proporção de 70/15/15, possibilitou a avaliação adequada do modelo em dados não vistos e o ajuste de seus hiperparâmetros.

Para otimizar o processo de treinamento, foi realizada uma amostragem aleatória do conjunto de treinamento, limitando o tamanho da amostra a 100.000 exemplos. Essa amostragem teve como objetivo reduzir o tempo de treinamento sem comprometer significativamente o desempenho do modelo.

A etapa de vetorização transformou os textos em sequências numéricas, utilizando um tokenizador da biblioteca Keras. O tokenizador foi instanciado com um vocabulário limitado às 50.000 palavras mais frequentes, reservando um token para palavras fora do vocabulário (OOV). O tokenizador foi ajustado aos textos do conjunto de treinamento amostrado e, em seguida, aplicado aos conjuntos de treinamento, validação e teste, convertendo os textos em sequências numéricas.

Para garantir que todas as sequências tivessem o mesmo comprimento, foi aplicado padding às sequências, utilizando um comprimento máximo de 100 e preenchimento à direita.

Os rótulos das classes foram codificados utilizando um LabelEncoder, ajustado aos rótulos do conjunto de treinamento amostrado. Os rótulos dos conjuntos de treinamento,

validação e teste foram, então, transformados em representações numéricas e, posteriormente, convertidos para codificação one-hot, preparando-os para o treinamento do modelo.

4 METODOLOGIA

A metodologia empregada neste trabalho foi estruturada em uma série de etapas para realizar a classificação de textos. Inicialmente, o conjunto de dados foi carregado e submetido a um pré-processamento detalhado, que compreendeu a remoção de valores ausentes, a análise e o tratamento do desbalanceamento de classes e a limpeza do texto. O conjunto de dados resultante foi então dividido em conjuntos de treinamento, validação e teste, permitindo a avaliação do desempenho do modelo em diferentes contextos. A vetorização dos textos foi realizada utilizando um tokenizador, seguido pela aplicação de padding para uniformizar o comprimento das sequências. Os rótulos das categorias foram codificados e preparados para o treinamento do modelo. Por fim, um modelo de rede neural convolucional (CNN) foi construído, treinado e avaliado em sua capacidade de classificar os textos nas categorias especificadas.

4.1 TÉCNICA UTILIZADA

A técnica central utilizada neste trabalho foi a rede neural convolucional (CNN). As CNNs, amplamente aplicadas em visão computacional, demonstraram sua eficácia em tarefas de Processamento de Linguagem Natural (PLN). Neste contexto, o modelo CNN foi projetado para extrair características discriminativas dos textos, possibilitando a classificação em categorias predefinidas. A arquitetura do modelo foi composta por uma camada de *embedding* para representar as palavras em um espaço vetorial, camadas convolucionais para identificar padrões locais, uma camada de *max pooling* global para reduzir a dimensionalidade, camadas densas para a classificação e camadas de *dropout* para regularização. O modelo foi compilado utilizando o otimizador Adam e a função de perda *categorical crossentropy*, adequada para problemas de classificação multiclasse.

4.2 EXPERIMENTO PARA AVALIAR A TÉCNICA UTILIZADA

O experimento para avaliar a eficácia da técnica utilizada envolveu o treinamento e a

avaliação do modelo CNN nos conjuntos de treinamento, validação e teste. O treinamento foi conduzido com um número predeterminado de épocas e tamanho de batch, empregando a técnica de early stopping para mitigar o overfitting. O desempenho do modelo foi avaliado utilizando a acurácia como métrica principal, calculada nos conjuntos de validação e teste. Adicionalmente, a perda (loss) foi monitorada durante o treinamento para acompanhar a convergência do modelo. A avaliação final do modelo no conjunto de teste forneceu uma estimativa do seu desempenho em dados não vistos, indicando sua capacidade de generalização.

5 RESULTADOS

A avaliação do modelo de classificação revelou um desempenho geral notavelmente elevado na tarefa de categorizar os textos médicos. A acurácia global do modelo no conjunto de teste alcançou 0.9797, demonstrando sua alta capacidade de classificar corretamente as amostras em suas respectivas especialidades. Adicionalmente, a perda (loss) no conjunto de teste foi de 0.0795, indicando um baixo nível de erro na classificação.

A análise detalhada das métricas de precisão, recall e F1-score para cada especialidade fornece uma visão mais aprofundada do desempenho do modelo. A precisão, que mede a proporção de previsões positivas corretas entre todas as instâncias classificadas como positivas, atingiu o valor máximo de 1.00 para a maioria das especialidades, incluindo "Colorectal Cancer", "Breast Cancer", "Lung Cancer", "Parkinson's Disease", "Gum (Periodontal) Disease", "Prostate Cancer", "Brody myopathy" e "High Blood Pressure". A exceção notável é a categoria "Other specialty", que apresentou uma precisão de 0.99, indicando uma excelente capacidade de classificar corretamente essa categoria também.

O recall, que quantifica a proporção de instâncias positivas corretamente identificadas pelo modelo, também alcançou 1.00 para a maioria das especialidades. No entanto, as categorias "Prostate Cancer" e "Diabetes" apresentaram um recall de 0.83, sugerindo que o modelo pode ter alguma dificuldade em identificar todas as instâncias relevantes dessas categorias.

A métrica F1-score, que representa a média harmônica entre precisão e recall, confirma o bom desempenho geral do modelo. A maioria das especialidades obteve um F1-score de 1.00, indicando um equilíbrio perfeito entre precisão e recall. As categorias "Prostate Cancer"

e "Diabetes" apresentaram um F1-score de 0.91, refletindo o impacto do recall ligeiramente inferior nessas categorias.

Em resumo, os resultados demonstram a eficácia do modelo na classificação das especialidades médicas, com alta acurácia global e bom desempenho nas métricas de precisão, recall e F1-score para a maioria das categorias(Figura 1). As categorias "Prostate Cancer" e "Diabetes" apresentaram um desempenho ligeiramente inferior em termos de recall, o que pode indicar a necessidade de investigação adicional para aprimorar a capacidade do modelo de identificar corretamente todas as instâncias dessas categorias.

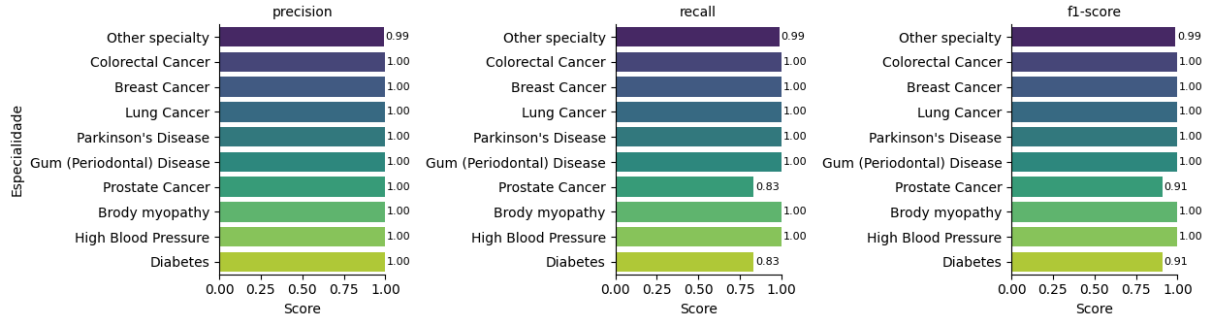

Figura 1

Além das métricas de avaliação no conjunto de teste, a análise das curvas de desempenho durante o treinamento oferece insights importantes sobre o processo de aprendizado do modelo. A Figura 2 apresenta as curvas de perda (loss) e acurácia para os conjuntos de treinamento e validação ao longo das épocas.

Observa-se que tanto a perda no conjunto de treinamento (train_loss) quanto a perda no conjunto de validação (val_loss) diminuem significativamente nas primeiras épocas, indicando que o modelo está aprendendo a classificar os textos com maior precisão. Após aproximadamente 10 épocas, a perda em ambos os conjuntos continua a diminuir, mas a taxa de diminuição se torna mais lenta, sugerindo que o modelo está se aproximando de um ponto de convergência. Nota-se, também, que as curvas de perda de treinamento e validação permanecem relativamente próximas ao longo do treinamento, o que sugere que o modelo não está sofrendo de overfitting significativo.

De forma semelhante, a acurácia nos conjuntos de treinamento (train_acc) e validação (val_acc) aumenta rapidamente nas primeiras épocas, indicando uma melhoria na capacidade do modelo de classificar corretamente os textos. Após cerca de 10 épocas, a acurácia em

ambos os conjuntos continua a aumentar, mas a taxa de aumento diminui, consistente com a observação feita nas curvas de perda. Novamente, as curvas de acurácia de treinamento e validação permanecem próximas, reforçando a conclusão de que o modelo generaliza bem para os dados de validação e não apresenta overfitting pronunciado.

Em conjunto com a alta acurácia obtida no conjunto de teste, a análise das curvas de desempenho durante o treinamento corrobora a eficácia do modelo na tarefa de classificação de textos médicos. A convergência das curvas de perda e acurácia, juntamente com a ausência de overfitting evidente, sugere que o modelo aprendeu padrões relevantes nos dados e generalizou bem para dados não vistos.

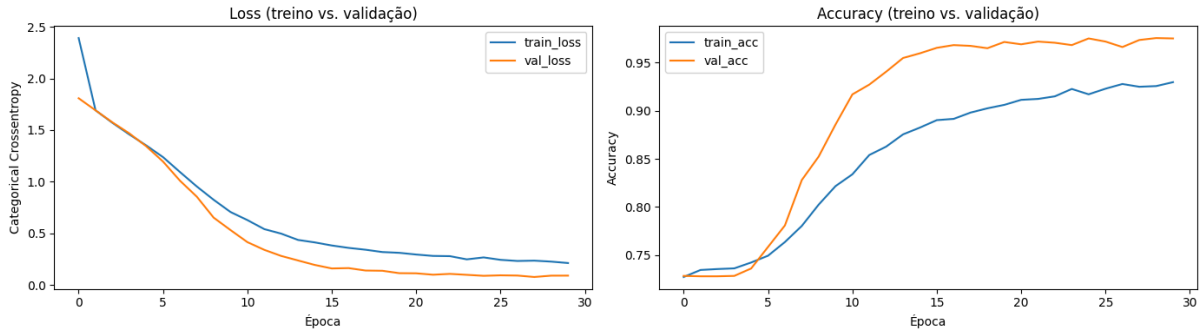

Figura 2

6 REFERÊNCIAS

ABACHA, A.; DEMNER-FUSHMAN, D. A Question-Entailment Approach to Question Answering. BMC Bioinformatics, v. 20, n. 1, p. 511, 2019. Disponível em: https://doi.org/10.1186/s12859-019-3119-4. Acesso em: 21 de Abril de 2025.