Relatório do Projeto: Sistema de Diagnóstico por Sintomas com PLN

Felipe Macêdo Dutra
felipe.dutra01@fatec.sp.gov.br
Faculdade de Tecnologia - Rubens Lara

1 Introdução

Sistemas de diagnóstico por sintomas desempenham um papel crucial na área da saúde, oferecendo suporte na identificação de possíveis condições médicas. Este projeto tem como objetivo desenvolver um sistema de diagnóstico que utiliza técnicas de *machine learning* para analisar a descrição dos sintomas fornecidos pelo usuário e identificar a doença mais provável.

1.1 Objetivos

O objetivo principal deste projeto é desenvolver um sistema de diagnóstico por sintomas que, a partir da entrada de sintomas em linguagem natural, forneça uma lista de possíveis doenças ordenadas por grau de similaridade.

Os objetivos específicos incluem:

- Achar um dataset abrangente de doenças e seus respectivos sintomas.
- Implementar um modelo de *machine learning* utilizando a técnica de similaridade do cosseno.
- Desenvolver uma interface gráfica amigável para facilitar a interação do usuário com o sistema.

2 Revisão Bibliográfica

2.1 Processamento de Linguagem Natural (PLN)

O Processamento de Linguagem Natural (PLN) é um campo da Inteligência Artificial que se dedica à capacidade dos computadores de entender, interpretar e gerar linguagem humana. No contexto deste projeto, o PLN é utilizado para processar a entrada de sintomas do usuário, realizando tarefas como tokenização (divisão do texto em palavras) e remoção de *stopwords* (palavras comuns que não agregam significado).

2.2 Similaridade do Cosseno

A similaridade do cosseno é uma medida de similaridade entre dois vetores em um espaço multidimensional. No sistema de diagnóstico, essa técnica é utilizada para calcular o grau de semelhança entre o vetor que representa os sintomas inseridos pelo usuário e os vetores que representam os sintomas associados a cada doença no dataset.

$$\text{similaridade do cosseno}(A, B) = \frac{A \cdot B}{||A|| \cdot ||B||}$$

Onde:

- $A \cdot B$: produto escalar dos vetores.
- $||A||$ e $||B||$: magnitudes dos vetores.

3 Materiais e Métodos

3.1 Dataset

O dataset utilizado é o "Symptom2Disease_translated.csv" foi um dataset original em inglês traduzido que contém informações sobre doenças e seus respectivos sintomas em português. O dataset contém 1200 linhas válidas, 24 doenças diferentes, cada doença tem 50 descrições de sintomas e possui as seguintes colunas:

- ID da doença: Identificador.
- Nome da doença: Nome da doença.
- Sintomas: Lista de sintomas associados à doença.

3.2 Bibliotecas Python

- `pandas`: Para manipulação e tratamento dos dados do dataset.
- `nltk` :Para processamento de linguagem natural (tokenização, remoção de *stopwords*).
- `scikit-learn` : Para a implementação do modelo de *machine learning* (`TfidfVectorizer`, `cosine_similarity`).
- `tkinter` : Para a criação da interface gráfica.

3.3 Implementação do Modelo

Os passos principais:

1. Pré-processamento dos dados.
2. Vetorização com `TfidfVectorizer`.

3. Cálculo da similaridade do cosseno.

4. Ordenação dos resultados.

4 Desenvolvimento do Sistema

4.1 Leitura e Pré-Processamento dos Dados

Listing 1: Função de tratamento de dados

```
def tratamento_dados(texto):
    texto = str(texto).lower()
    texto = texto.translate(str.maketrans("", "", string.
        punctuation))
    tokens = tokenizer.tokenize(texto)
    stop_words = set(stopwords.words('portuguese'))
    tokens = [palavra for palavra in tokens if palavra not
        in stop_words]
    return ''.join(tokens)
```

4.2 Implementação da Recomendação de Doenças

Listing 2: Função de recomendação de doenças

```
def recomenda_top_doencas(descricao_usuario, df_base, top_n
    =3):
    # Pré-processa os sintomas digitados pelo usuário
    consulta_tratada = tratamento_dados(descricao_usuario)

    # Cria um corpus com os sintomas do dataset e a nova
        consulta
    corpus = df_base['sintomas_tratados'].tolist() + [
        consulta_tratada]

    # Vetoriza os textos com tf-idf
    tfidf = TfidfVectorizer()
    tfidf_matriz = tfidf.fit_transform(corpus)

    # Calcula a similaridade do cosseno entre a consulta (ú
        ltimo vetor) e os demais
    # tfidf_matriz[-1] representa o vetor da descrição do
        usuário
    # tfidf_matriz[:-1] representa os vetores das doenças no
         dataset
    cosine_sim = cosine_similarity(tfidf_matriz[-1],
        tfidf_matriz[:-1]).flatten()

    resultados_temp = []
```

```
# Associa cada similaridade ao nome da doença
for idx, score in enumerate(cosine_sim):
    doenca = df_base.iloc[idx]['Doenca']
    resultados_temp.append((doenca, score))

# Garante que apenas o maior score por doença seja
    considerado
resultados_dict = {}
for doenca, score in resultados_temp:
    if doenca not in resultados_dict or score >
        resultados_dict[doenca]:
        resultados_dict[doenca] = score

# Ordena as doenças por maior similaridade
resultados_ordenados = sorted(resultados_dict.items(),
    key=lambda x: x[1], reverse=True)

# Retorna resultados como percentual
resultados_finais = [(doenca, sim * 100) for doenca, sim
    in resultados_ordenados[:top_n]]
return resultados_finais
```

4.3 Interface Gráfica

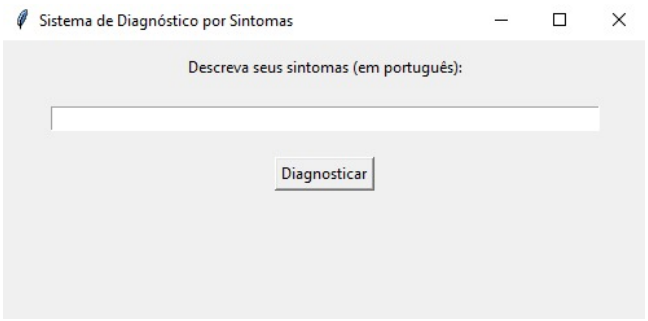

Figure 1: Interface Gráfica do Sistema de Diagnóstico por Sintomas

5 Resultados

5.1 Exemplos de Uso

Exemplo 1: Entrada: "dor de barriga e febre" , Saída:

- Tifóide (47,55%)
- Malária (22,13%)

- Dengue (15,33%)

EXEMPLO ENCONTRADO NO DATASET (102, tifóide: "Eu tenho sofrido calafrios e febre, juntamente com dor abdominal intensa. Estou me sentindo realmente infeliz no geral, e eu simplesmente não consigo abalar esses sintomas")
Exemplo 2: Entrada: "dor no peito, falta de ar" , Saída:

- Pneumonia (33,36%)
- Hipertensão (32,49%)
- Asma Brônquica (24,79%)

EXEMPLO ENCONTRADO NO DATASET (146, pneumonia, "Estou tendo muitos problemas para respirar. Não estou me sentindo bem e estou suando muito. Tenho muito muco na garganta e meu peito dói. Minha respiração é trabalhada e a fleuma que estou tossindo tem uma tonalidade estranha")

6 Conclusão

Este projeto demonstrou a viabilidade de desenvolver um sistema de diagnóstico por sintomas utilizando uma técnica de *machine learning*, o PLN. O sistema implementado é capaz de fornecer diagnósticos preliminares com base na descrição dos sintomas do usuário.

Mas é válido ressaltar que a precisão da taxa de acerto da doença é totalmente dependente do dataset, ou seja, quanto maior o número de informações no dataset maior a precisão.