# Tarefa de programação: Programa completo - Similaridades entre textos - Caso COH-PIAH

Atividade não enviada. Você deve obter 8/10 pontos para passar.

**Prazo** Seja aprovado nessa tarefa até December 16, 10:59 PM PST

**Instruções**

Meus envios

Discussões

## Introdução

John é monitor na matéria de Introdução à Produção Textual I na Penn State University (PSU). Durante esse período, John descobriu que uma epidemia de COH-PIAH estava se espalhando pela PSU. Essa doença rara e altamente contagiosa faz com que as pessoas contaminadas produzam textos extremamente semelhantes de forma involuntária. Após a entrega da primeira redação, John desconfiou que alguns alunos estavam sofrendo de COH-PIAH. John, se preocupando com a saúde da turma, resolveu buscar um método para identificar os casos de COH-PIAH. Para isso, ele necessita da sua ajuda para desenvolver um programa que o auxilie a identificar os alunos contaminados.

## Detecção de autoria

Utilizando diferentes estatísticas do texto, é possível identificar aspectos que funcionam como uma “assinatura” do autor. Diferentes pessoas possuem diferentes estilos de escrita, algumas preferindo sentenças mais curtas, outras preferindo sentenças mais longas.

Essas “assinatura” pode ser utilizada para detecção de plágio, evidência forense, ou nesse caso, para detectar a grave doença COH-PIAH.

## Traços linguísticos

Nesse exercício utilizaremos as seguintes estatísticas para detectar a doença:

- Tamanho médio de palavra: Média simples do número de caracteres por palavra.
- Relação Type-Token: Número de palavras diferentes utilizadas em um texto divididas pelo total de palavras.
- Razão Hapax Legomana: Número de palavras utilizadas uma vez dividido pelo número total de palavras.

- Tamanho médio de sentença: Média simples do número de caracteres por sentença.
- Complexidade de sentença: Média simples do número de frases por sentença.
- Tamanho médio de frase: Média simples do número de caracteres por frase.

## Funcionamento do programa

Diversos estudos foram compilados e hoje se conhece precisamente a assinatura de um portador de COH-PIAH. Seu programa deverá receber diversos textos e calcular os valores dos diferentes traços linguísticos da seguinte forma:

- Tamanho médio de palavra é a soma dos tamanhos das palavras dividida pelo número total de palavras.
- Relação Type-Token é o número de palavras diferentes dividido pelo número total de palavras. Por exemplo, na frase "O gato caçava o rato", temos 5 palavras no total (o, gato, caçava, o, rato) mas somente 4 diferentes (o, gato, caçava, rato). Nessa frase, a relação Type-Token vale $\frac{4}{5} = 0.8$
- Razão Hapax Legomana é o número de palavras que aparecem uma única vez dividido pelo total de palavras. Por exemplo, na frase "O gato caçava o rato", temos 5 palavras no total (o, gato, caçava, o, rato) mas somente 3 que aparecem só uma vez (gato, caçava, rato). Nessa frase, a relação Hapax Legomana vale $\frac{3}{5} = 0.6$
- Tamanho médio de sentença é a soma dos números de caracteres em todas as sentenças dividida pelo número de sentenças (os caracteres que separam uma sentença da outra **não** devem ser contabilizados como parte da sentença).
- Complexidade de sentença é o número total de frases divido pelo número de sentenças.
- Tamanho médio de frase é a soma do número de caracteres em cada frase dividida pelo número de frases no texto (os caracteres que separam uma frase da outra **não** devem ser contabilizados como parte da frase).

Após calcular esses valores para cada texto, você deve comparar com a assinatura fornecida para os infectados por COH-PIAH. O grau de similaridade entre dois textos, $a$ e $b$, é dado pela fórmula:

$$S_{ab} = \frac{\sum_{i=1}^{6} ||f_{i,a} - f_{i,b}||}{6}$$

Onde:

- $S_{ab}$ é o grau de similaridade entre os textos $a$ e $b$;
- $f_{i,a}$ é o valor de cada traço linguístico $i$ no texto $a$; e
- $f_{i,b}$ é o valor de cada traço linguístico $i$ no texto $b$.

Perceba que quanto mais similares $a$ e $b$ forem, menor $S_{ab}$ será. Para cada texto, você deve calcular o grau de similaridade com a assinatura do portador de COH-PIAH e no final exibir qual o texto que mais provavelmente foi escrito por algum aluno infectado.

Exemplo:

```
$ > python3 coh_piah.py

Bem-vindo ao detector automático de COH-PIAH.


Entre o tamanho medio de palavra: 4.79
Entre a relação Type-Token: 0.72
Entre a Razão Hapax Legomana: 0.56
Entre o tamanho médio de sentença: 80.5
Entre a complexidade média da sentença: 2.5
Entre o tamanho medio de frase: 31.6

Digite o texto 1 (aperte enter para sair): Navegadores antigos tinham
uma frase gloriosa:"Navegar é preciso; viver não é preciso". Quero para
mim o espírito [d]esta frase, transformada a forma para a casar como eu
sou: Viver não é necessário; o que é necessário é criar. Não conto
gozar a minha vida; nem em gozá-la penso. Só quero torná-la
grande,ainda que para isso tenha de ser o meu corpo e a (minha alma) a
lenha desse fogo. Só quero torná-la de toda a humanidade;ainda que para
isso tenha de a perder como minha. Cada vez mais assim penso.Cada vez
mais ponho da essência anímica do meu sangueo propósito impessoal de
engrandecer a pátria e contribuirpara a evolução da humanidade.É a
forma que em mim tomou o misticismo da nossa Raça.

Digite o texto 2 (aperte enter para sair): Voltei-me para ela; Capitu
tinha os olhos no chão. Ergueu-os logo, devagar, e ficamos a olhar um
para o outro... Confissão de crianças, tu valias bem duas ou três
páginas, mas quero ser poupado. Em verdade, não falamos nada; o muro
falou por nós. Não nos movemos, as mãos é que se estenderam pouco a
pouco, todas quatro, pegando-se, apertando-se, fundindo-se. Não marquei
a hora exata daquele gesto. Devia tê-la marcado; sinto a falta de uma
nota escrita naquela mesma noite, e que eu poria aqui com os erros de
ortografia que trouxesse, mas não traria nenhum, tal era a diferença
entre o estudante e o adolescente. Conhecia as regras do escrever, sem
suspeitar as do amar; tinha orgias de latim e era virgem de mulheres.

Digite o texto 3 (aperte enter para sair): NOSSA alegria diante dum
sistema metafisico, nossa satisfação em presença duma construção do
pensamento, em que a organização espiritual do mundo se mostra num
conjunto lógico, coerente a harmônico, sempre dependem eminentemente da
estética; têm a mesma origem que o prazer, que a alta satisfação,
sempre serena afinal, que a atividade artística nos proporciona quando
cria a ordem e a forma a nos permite abranger com a vista o caos da
vida, dando-lhe transparência.

Digite o texto 4 (aperte enter para sair):

O autor do texto 2 está infectado com COH-PIAH
```

## Funções de suporte

As seguintes funções **devem** ser utilizadas no seu programa; algumas já estão implementadas, outras devem ser implementadas por você. Sinta-se livre para criar funções adicionais, caso necessário. Utilize este esqueleto como base para começar o seu programa.

**Dica:** aproveite as funções pré-prontas do esqueleto, como "separa_sentenca", "separa_frase" etc.! Como há mais de uma maneira de pensar a separação entre frases/palavras/sentenças, usando essas funções você vai fazer o cálculo da maneira esperada pelo corretor automático.

**Cuidado:** A função le_textos() considera que um "texto" é uma linha de texto, ou seja, não é possível inserir parágrafos separados. Se você digitar algum "enter", a função vai entender que você está começando um novo texto. Preste especial atenção a isso se usar "copiar/colar" para inserir os textos! Note também que, no cálculo de similaridade, é preciso encontrar o valor absoluto de cada uma das diferenças.

```
import re

def le_assinatura():
  '''A funcao le os valores dos tracos linguisticos do modelo e devolve
       uma assinatura a ser comparada com os textos fornecidos'''
    print("Bem-vindo ao detector automático de COH-PIAH.")

    wal = float(input("Entre o tamanho medio de palavra:"))
    ttr = float(input("Entre a relação Type-Token:"))
    hlr = float(input("Entre a Razão Hapax Legomana:"))
    sal = float(input("Entre o tamanho médio de sentença:"))
    sac = float(input("Entre a complexidade média da sentença:"))
    pal = float(input("Entre o tamanho medio de frase:"))

    return [wal, ttr, hlr, sal, sac, pal]

def le_textos():
    i = 1
    textos = []
    texto = input("Digite o texto " + str(i) +" (aperte enter para sair
        ):")
    while texto:
        textos.append(texto)
        i += 1
        texto = input("Digite o texto " + str(i) +" (aperte enter para
            sair):")

    return textos

def separa_sentencas(texto):
    '''A funcao recebe um texto e devolve uma lista das sentencas
        dentro do texto'''
    sentencas = re.split(r'[.!?]+', texto)
    if sentencas[-1] == '':
        del sentencas[-1]
    return sentencas

def separa_frases(sentenca):
    '''A funcao recebe uma sentenca e devolve uma lista das frases
        dentro da sentenca'''
    return re.split(r'[,:;]+', sentenca)

def separa_palavras(frase):
    '''A funcao recebe uma frase e devolve uma lista das palavras
        dentro da frase'''
    return frase.split()

def n_palavras_unicas(lista_palavras):
    '''Essa funcao recebe uma lista de palavras e devolve o numero de
        palavras que aparecem uma unica vez'''
    freq = dict()
    unicas = 0
    for palavra in lista_palavras:
        p = palavra.lower()
        if p in freq:
            if freq[p] == 1:
                unicas -= 1
            freq[p] += 1
        else:
            freq[p] = 1
            unicas += 1

    return unicas

def n_palavras_diferentes(lista_palavras):
    '''Essa funcao recebe uma lista de palavras e devolve o numero de
        palavras diferentes utilizadas'''
    freq = dict()
    for palavra in lista_palavras:
        p = palavra.lower()
        if p in freq:
            freq[p] += 1
        else:
            freq[p] = 1

    return len(freq)

def compara_assinatura(as_a, as_b):
```

```
71      '''IMPLEMENTAR. Essa funcao recebe duas assinaturas de texto e deve
            devolver o grau de similaridade nas assinaturas.'''
72      pass
73
74  def calcula_assinatura(texto):
75      '''IMPLEMENTAR. Essa funcao recebe um texto e deve devolver a
            assinatura do texto.'''
76      pass
77
78  def avalia_textos(textos, ass_cp):
79      '''IMPLEMENTAR. Essa funcao recebe uma lista de textos e deve
            devolver o numero (1 a n) do texto com maior probabilidade de
            ter sido infectado por COH-PIAH.'''
80      pass
```

## Exemplo de Assinatura

Um passo importante para seu programa é calcular a assinatura dos textos corretamente. Para testar se sua função $calcula_assinatura$ está correta, deixamos aqui um exemplo de execução:

```
1  texto = "Muito além, nos confins inexplorados da região mais brega da
       Borda Ocidental desta Galáxia, há um pequeno sol amarelo e esquecido
       . Girando em torno deste sol, a uma distancia de cerca de 148
       milhões de quilômetros, há um planetinha verde-azulado absolutamente
       insignificante, cujas formas de vida, descendentes de primatas, são
       tão extraordinariamente primitivas que ainda acham que relógios
       digitais são uma grande ideia."
2  calcula_assinatura(texto)
3  >[5.571428571428571, 0.8253968253968254, 0.6984126984126984, 210.0, 4.5,
       45.888888888888886]
4
5
```

## How to submit

When you're ready to submit, you can upload files for each part of the assignment on the "My submission" tab.