**Introdução à Ciência da Computação com Python (Parte 1) – USP (Coursera)**

**SEMANA 9 - Tarefa de programação: Programa completo - Similaridades entre textos - Caso COH-PIAH**

**Introdução**

Manuel Estandarte é monitor na disciplina *Introdução à Produção Textual I* na Universidade de Pasárgada (UPA). Durante o período letivo, Manuel descobriu que uma epidemia de COH-PIAH estava se espalhando pela UPA. Essa doença rara e altamente contagiosa faz com que indivíduos contaminados produzam involuntariamente textos muito semelhantes aos de outras pessoas. Após a entrega da primeira redação, Manuel desconfiou que alguns alunos estavam sofrendo de COH-PIAH. Manuel, preocupado 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.

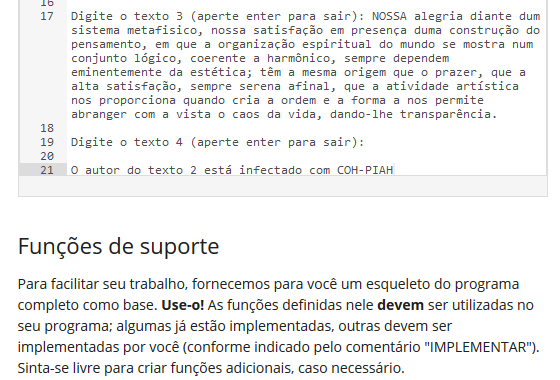
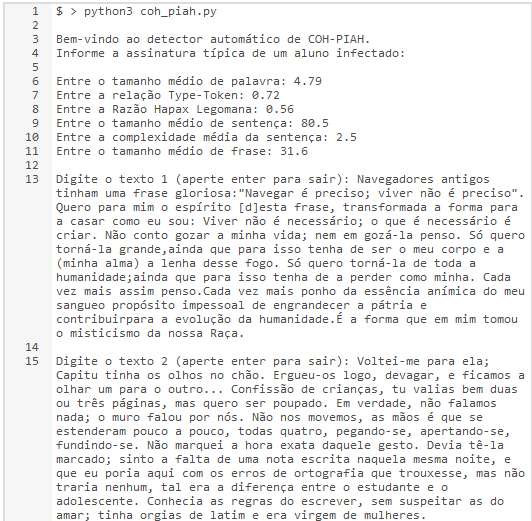
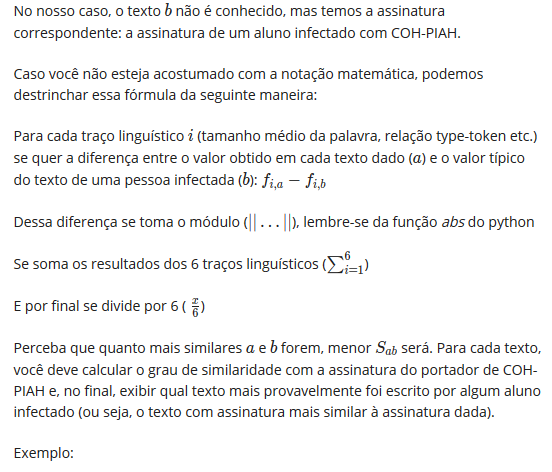
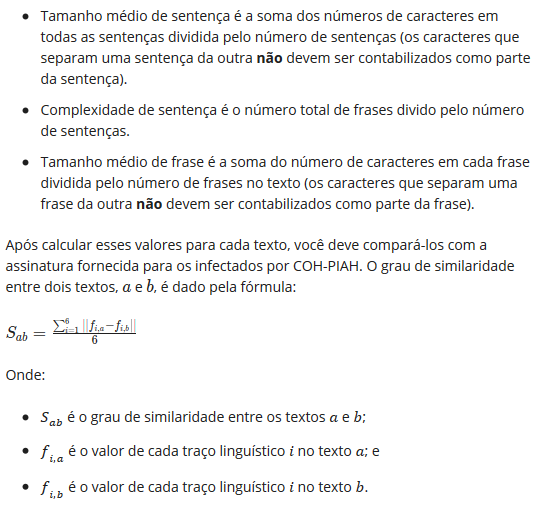
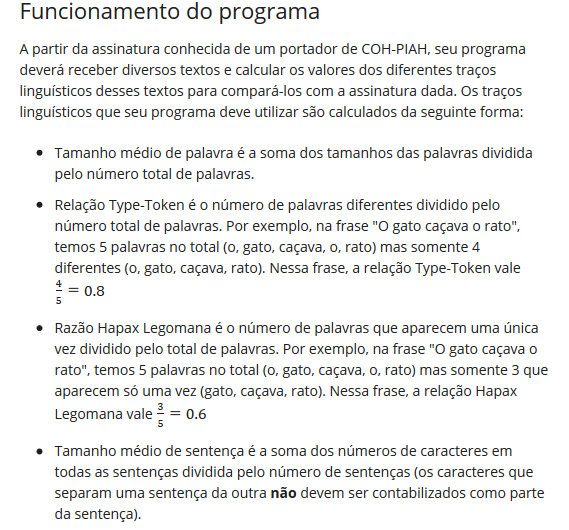
**Detecção de autoria**

Diferentes pessoas possuem diferentes estilos de escrita; por exemplo, algumas pessoas preferem sentenças mais curtas, outras preferem sentenças mais longas. Utilizando diversas estatísticas do texto, é possível identificar aspectos que funcionam como uma “assinatura” do seu autor e, portanto, é possível detectar se dois textos dados foram escritos por uma mesma pessoa. Ou seja, essa “assinatura” pode ser utilizada para detecção de plágio, evidência forense ou, neste caso, para diagnosticar a grave doença COH-PIAH.

**Traços linguísticos**

Neste 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.



Arquivo similaridades-textos.py (no Sublime)

**Dica:** não se preocupe com os detalhes de implementação das funções pré-prontas do esqueleto, como "separa\_sentenca()", "separa\_frase()" etc. nem com as definições exatas de frase e sentença. Essas funções já cuidam disso para você, e podem ser pensadas como "caixas pretas": você pode utilizá-las sabendo o que recebem e o que devolvem, mas não é necessário saber sobre os seus detalhes internos. Além de isso ser muito comum ao programar em equipe, 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.

**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:

