



78%

ATIVIDADES
2 de 7

DISCORD
ALURA

FÓRUM DO
CURSO

VOLTAR
PARA
DASHBOARD

MODO
NOTURNO



17.3k xp

a

02

Palavras desconhecidas ao vocabulário

Transcrição

Nosso corretor já é capaz de acertar aproximadamente 77% das palavras.

A taxa de erros está associada a dois fatores; **restrição do algoritmo** que não gera a palavra correta e **dados de treinamento**.

Às vezes um determinado termo ainda é **desconhecido**, e portanto não podemos realizar a correção de algo que não conhecemos.

Em um exemplo dado anteriormente, se conhecemos cerca de apenas 100 palavras de um idioma estrangeiro, somente poderemos corrigir esta mesma quantidade de palavras conhecidas.

Logo, o modelo do corretor é restrito por este volume de vocabulário, calculado em 17.654 vocábulos em nosso caso.

Precisaremos calcular o quanto que essa restrição está impactando no resultado, pois temos uma taxa de erro e deveremos saber o quanto desta porcentagem está associada às palavras desconhecidas.



78%

ATIVIDADES
2 de 7

DISCORD
ALURA

FÓRUM DO
CURSO

VOLTAR
PARA
DASHBOARD

MODO
NOTURNO



17.3k xp

a

Para isso, usaremos a função `avaliador()`. Copiaremos seu bloco de código e colaremos em uma nova célula seguinte no Google Colab.

```
def avaliador(testes):  
    numero_palavras = len(testes)  
    acertou = 0  
    for correta, errada in testes:  
        palavra_corrigida = corretor(errada)  
        if palavra_corrigida == correta:  
            acertou += 1  
    taxa_acerto = round(acertou*100/numero_palavras, 2)  
    print(f"{taxa_acerto}% de {numero_palavras} palavras")
```

`avaliador(lista_teste)`

COPIAR CÓDIGO

Como estamos calculando a nossa taxa de acerto, poderemos calcular também a taxa de palavras desconhecidas por meio de uma nova variável chamada `taxa_desconhecida`. Esta será igual a algo muito parecido com o cálculo da taxa de acerto.

Porém, não queremos mais calcular a `acertou`, e sim o que é desconhecido; então substituiremos pela nova variável `desconhecida`. Na função, *setaremos* seu valor com um contador de palavras desconhecidas, da mesma forma que fizemos com `acertou` igual a `0`.

Incrementaremos este contador a cada vez que descobirmos que uma das palavras corretas em `correta` é desconhecida ao nosso vocabulário.



78%

ATIVIDADES
2 de 7

DISCORD
ALURA

FÓRUM DO
CURSO

VOLTAR
PARA
DASHBOARD

MODOS
NOTURNO



17.3k xp

a

Esta implementação entrará no `if` se acertou o termo. Já o `else` será justamente quando errar a previsão, ou seja, incrementaremos a variável `desconhecida` quando a palavra correta for desconhecida do meu vocabulário. Se errou, poderá ser por algum dos dois motivos já citados.

Após `+=`, será incrementada quando a palavra correta não estiver no `vocabulario` com `not in` entre parênteses, e precisaremos recebê-lo como parâmetro de `avaliador()` também.

Com isso, recebemos o `vocabulario` como parâmetro do avaliador, incrementamos a variável `desconhecida` quando a palavra correta não estiver no meu vocabulário e calculamos a taxa de `taxa_desconhecida`

Falta-nos imprimi-la também na frase e *setar* a variável `vocabulario` sendo igual a `set()` recebendo a `lista_normalizada`.

Ao final do código, passaremos `vocabulario` como segundo parâmetro da chamada de `avaliador()`.

```
def avaliador(testes, vocabulario):  
    numero_palavras = len(testes)  
    acertou = 0  
    desconhecida = 0  
    for correta, errada in testes:  
        palavra_corrigida = corretor(errada)  
        if palavra_corrigida == correta:  
            acertou += 1  
    else:
```



78%

ATIVIDADES
2 de 7

DISCORD
ALURA

FÓRUM DO
CURSO

VOLTAR
PARA
DASHBOARD

MODO
NOTURNO



17.3k xp

a

```
desconhecida += (correta not in vocabulario)

taxa_acerto = round(acertou*100/numero_palavras, 2)
taxa_desconhecida = round(desconhecida*100/numero_palavras, 2)
print(f"{taxa_acerto}% de {numero_palavras} palavras, desconhecidas é {taxa_
```

```
vocabulario = set(lista_normalizada)
avaliador(lista_teste, vocabulario)
```

COPIAR CÓDIGO



Como retorno da execução, teremos a taxa de acerto de 76,34% e a taxa de palavras desconhecidas de 6.99% .

A seguir, descobriremos se há um maneira de tornarmos nosso algoritmo **mais abrangente** com alguma expansão de metodologia.