



90%

ATIVIDADES
3 de 8

DISCORD
ALURA

FÓRUM DO
CURSO

VOLTAR
PARA
DASHBOARD

MODO
NOTURNO



17.5k xp

a

03

Avaliando o resultado dos dois corretores cont

Transcrição

Vimos que o `novo_corretor()` performou consideravelmente pior do que o nosso corretor anterior. Printaremos alguns resultados para analisarmos o porquê deste novo não estar indo tão bem quanto o outro.

Logo após o `if`, colocaremos um `else` para toda vez em que errarmos a palavra a ser corrigida. Imprimiremos a palavra errada e concatenaremos com um hífen para separarmos da palavra errada gerada pelo antigo `corretor()`.

Em seguida, concatenaremos da mesma forma com a palavra corrigida pelo `novo_corretor()`.

Imprimiremos a equivocada, a corrigida pelo meu antigo corretor e a corrigida pelo meu novo corretor para analisarmos os resultados.

```
def avaliador(testes, vocabulario):  
    numero_palavras = len(testes)  
    acertou = 0  
    desconhecida = 0  
    for correta, errada in testes:
```



90%

ATIVIDADES
3 de 8

DISCORD
ALURA

FÓRUM DO
CURSO

VOLTAR
PARA
DASHBOARD

MODOS
NOTURNOS



17.5k xp

a

```
palavra_corrigida = novo_corretor(errada)
desconhecida += (correta not in vocabulario)
if palavra_corrigida == correta:
    acertou += 1
else:
    print(errada + "-" + corretor(errada) + "-" + palavra_corrigida)
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 {taxa_de

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

COPIAR CÓDIGO

Com a execução, veremos uma lista em que a primeira palavra é a errada, a segunda é a gerada pelo corretor antigo, e a terceira pelo nosso novo.

Na primeira linha, teremos a palavra digitada equivocadamente `esje`, onde provavelmente a pessoa substituiu o “j” pelo “s”. Assim, gerou as palavras certas `esse` e `se`.

Na terceira linha do resultado, teremos a palavra `dosa` onde digitaram um “a” a mais, pois está ao lado da tecla “s”.

Nosso corretor antigo fez a correção para `dos` corretamente. Porém, o novo corretor corrige para `do`, o que está incorreto, pois `novo_corretor` gera ambos os resultados.



90%

ATIVIDADES
3 de 8

DISCORD
ALURA

FÓRUM DO
CURSO

VOLTAR
PARA
DASHBOARD

MODO
NOTURNO



17.5k xp

a

Porém, `do` aparece muito mais vezes do que a palavra `dos`. Como dependemos dessa máxima probabilidade, a função exibirá o termo que possui uma maior chance de estar correta, mesmo que a outra palavra também seja uma possibilidade correta.

Muitas vezes nossos erros de digitação estão apenas a uma distância da palavra equivocada, ou seja, é mais comum esquecermos de digitar ou trocarmos uma única letra.

Embora teoricamente o `novo_corretor()` parecesse ser melhor, está performando bem pior do que o outro. Portanto, é mais interessante continuarmos com o antigo.

Modificaremos o `avaliador()` novamente. Copiaremos seu bloco e o colaremos em uma nova célula, e ao invés de chamarmos o novo corretor, o deletaremos e chamaremos o antigo `corretor()`. O `print()` em `else` também não nos interessa mais, então poderemos apagá-lo.

```
def avaliador(testes, vocabulario):
    numero_palavras = len(testes)
    acertou = 0
    desconhecida = 0
    for correta, errada in testes:
        palavra_corrigida = corretor(errada)
        desconhecida += (correta not in vocabulario)
        if palavra_corrigida == correta:
            acertou += 1
    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 {taxa_de
```



90%

ATIVIDADES
3 de 8

DISCORD
ALURA

FÓRUM DO
CURSO

VOLTAR
PARA
DASHBOARD

MODOS
NOTURNOS



17.5k xp

a

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

COPIAR CÓDIGO

Rodando este código para verificarmos o resultado final do `avaliador()`, veremos nosso corretor tem 76.34% e desconhece aproximadamente 7% das palavras. Portanto, possui uma performance bem interessante.

Para finalizarmos, teremos uma nova célula para analisarmos seu funcionamento com outros exemplos de termos errados.

Imprimiremos o `novo_corretor()` e o antigo corretor para os nossos dados. Como exemplo, teremos a palavra sendo igual a "lgica".

```
palavra = "lgica"
print(novo_corretor(palavra))
print(corretor(palavra))
```

COPIAR CÓDIGO

O novo corretor acaba corrigindo para `fica`, e o nosso corretor antigo e ideal para esses nossos dados fez a correção correta.

Já no caso da palavra "lógica", os dois retornam a palavra correta.



90%

ATIVIDADES
3 de 8

DISCORD
ALURA

FÓRUM DO
CURSO

VOLTAR
PARA
DASHBOARD

MODO
NOTURNO



17.5k xp

a

Se eu colocarmos mais um “i” equivocadamente obtendo "lóiigica" , veremos que nosso corretor definitivo não conseguirá corrigir esse tipo de erro.

Como retorno, veremos um alóiigica e a palavra correta lógica .

Poderemos testar mais e verificar que nosso algoritmo de corretor é eficiente e corrige muitas palavras.

Para melhorarmos ainda mais, poderemos procurar uma base de dados maior para fazermos esse treinamento e termos uma performance ainda melhor.