



0%

ATIVIDADES
1 de 8DISCORD
ALURAFÓRUM DO
CURSOVOLTAR
PARA
DASHBOARDMODO
NOTURNO

15.8k xp



Transcrição

Boas vindas a mais um curso de **Processamento de Linguagem Natural** aqui na **Plataforma Alura!**

Sou o instrutor **Thiago Santos**.

Em nossas aulas, construiremos um **corretor ortográfico** em português utilizando **Python** e aplicando técnicas de **NLP**.

Além disso, discutiremos diversos assuntos como a comunicação entre máquina e humano por exemplo, e verificaremos que não é feita de maneira direta, ou seja, há um **intermediário** que é justamente o Processamento de Linguagem Natural ou a sigla em inglês NLP.

Esta técnica tem diversas aplicações como os Assistentes Pessoais por exemplo; o **Google Assistant**, o **Siri** da **Apple**, **Alexa** e etc.

Também faremos a **Análise de Sentimento**; no caso da opinião de uma pessoa sobre um filme, conseguiremos avaliá-la como positiva ou negativa, os tradutores, e até o buscador do **Google** utiliza bastante estes recursos, bem como o corretor ortográfico que construiremos.

Para entendermos melhor sua utilidade, imaginemos a seguinte situação: temos uma palavra digitada equivocadamente, como “lgica” por exemplo. Precisaremos passá-la pelos algoritmos de

NLP, o qual gerará e retornará a palavra correta, que neste caso é “lógica”.

Para criarmos o corretor, precisaremos de uma **base de treinamento** com um compilado de todos os artigos da Alura que temos. Porém, também avaliaremos se esta é adequada de fato.

Portanto, discutiremos as **métricas** que podem nos ajudar a descobrir se a base de dados é boa o suficiente para treinarmos o nosso modelo. Para isso, separaremos o *corpus* cheio de palavras de nosso texto em pequenas unidades chamados *tokens*.

Utilizaremos o `nltk` para realizar o processo de separação do texto em pequenas `strings` conhecido como **tokenização**. Em seguida, separaremos apenas as palavras desse tokens, formando uma lista.

Com isso, teremos uma quantidade e verificaremos quantos termos nosso algoritmo aprenderá de fato. Também normalizaremos o nosso texto, pois ainda possui palavras maiúsculas e minúsculas.

Em seguida, iniciaremos a construção do nosso corretor; iniciaremos inserindo letras, e sem querer digitamos uma letra a menos, então precisaremos inserir uma letra a mais.

Depois que construirmos esse algoritmo capaz de gerar as palavras com uma letra a mais, avaliaremos o nosso resultado.

Precisaremos de uma **base de teste**, que em nosso caso é a `palavras.txt` , a qual contém a palavra digitada equivocadamente e a correta.

Feita a avaliação, chegaremos à conclusão de que apenas nosso algoritmo com uma letra a mais não está com o resultado muito bom, pois está corrigindo apenas 1% das palavras. Logo, precisaremos tornar o nosso algoritmo **mais abrangente**.



0%

ATIVIDADES
1 de 8

DISCORD
ALURA

FÓRUM DO
CURSO

VOLTAR
PARA
DASHBOARD

MODOS
NOTURNO



15.8k xp

a

Para isso, implementaremos diversos outros algoritmos, como por exemplo

`deletando_caracteres()` , `troca_letra()` , `inverte_letra()` , e assim sucessivamente até chegarmos em um resultado satisfatório de aproximadamente 77% de acerto.

Depois, testaremos um `gerador_turbinado()` e veremos sua performance. Tiraremos conclusões e compararemos os dois corretores, mesmo que este seja aparentemente muito mais robusto.

Ao final, realizaremos alguns testes para digitarmos uma palavra equivocadamente e verificarmos a saída.

Vamos lá!



0%

ATIVIDADES
1 de 8

DISCORD
ALURA

FÓRUM DO
CURSO

VOLTAR
PARA
DASHBOARD

MODO
NOTURNO



15.8k xp

a